Ann Appl Sport Sci, 11(s1): e1222.
http://www.aassjournal.com; e-ISSN: 2322–4479; p-ISSN: 2476–4981. 10.61186/aassjournal.1222

ORIGINAL ARTICLE

Fraud Detection in Enterprise Resource Planning Systems Using One-Class Support Vector Machine Combined with Convolutional Neural Network: The Case of *Spor Istanbul*

[1]**Emrah Arslan***, [1]**Ali Güneş**

[1]Department of Computer Engineering, Faculty of Engineering, Istanbul Aydın University, Istanbul, Turkey.

Submitted April 31, 2023; Accepted in final form June 24, 2023.

ABSTRACT

Background. Combining a One-Class Support Vector Machine (OCSVM) with a Convolutional Neural Network (CNN) is presented as a novel technique for detecting fraud in Enterprise Resource Planning (ERP) systems. **Objectives.** The objective of this research is to develop a technique for detecting fraud in Enterprise Resource Planning (ERP) systems by combining a One-Class Support Vector Machine (OCSVM) with a Convolutional Neural Network (CNN), suitable for the Spor Istanbul ERP system. **Methods.** This study examines the ERP system utilized by Spor Istanbul, the largest sports enterprise in Turkey, as a case study. The study utilizes a custom database of web-based program files to create a dataset of benign and malicious JavaScript applications. Firstly, the text and control flow graph of the program is analyzed. Secondly, the OCSVM method is applied as an outlier detection technique, and CNN is used as a classifier. **Results.** The experimental results indicate that the proposed OCSVM-CNN approach achieves higher accuracy (96.78%) in detecting malicious scripts compared to using only CNN (94.8%). **Conclusion.** The research contributes to the development of multi-layered ERP software architecture with AI decision support, improving fraud detection in ERP systems.

KEYWORDS: ***Enterprise Resource Planning, Intelligent ERP, Forecasting, Demand Prediction, Decision Support System, Artificial Neural Networks, Cellular Neural Networks, Multi-Layered Software Architecture.***

INTRODUCTION

The use of One-Class Support Vector Machine (OC-SVM) combined with Convolutional Neural Networks (CNNs) has emerged as an effective solution for real-time fraud detection in Enterprise Resource Planning (ERP) systems (1). Traditional rule-based methods of fraud detection are often time-consuming, require expert knowledge, and may fail to detect newer types of fraud. In contrast, the combination of OC-SVM and CNNs allows for the identification of unusual activities that deviates from normal patterns by analyzing data patterns. Having a robust financial management system, including efficient fraud detection, is crucial for businesses to prevent financial loss and protect their reputation.

Enterprise Resource Planning (ERP) systems play a fundamental role in financial management in the information age (2). These systems integrate an enterprise's production and financial management, enabling the finance department to promptly obtain accurate business data. By improving the management level of enterprise finance, ERP systems promote the rational use of funds and reduce the risk of decision-making errors.

*. Corresponding Author:
Emrah Arslan, Ph.D.
E-mail: emraha@stu.aydin.edu.tr

While nonlinear forecasting methods such as Support Vector Machine (SVM) and Random Forest (RF) have been applied to economic forecasting in the context of ERP risk forecasting, these methods often struggle to capture the nonlinear relationships inherent in ERP risks, limiting the accuracy of risk prediction (3). Deep learning, on the other hand, has shown promise in extracting complex and effective features by learning through multiple networks. While deep learning has been successfully applied in predicting stock trading and daily closing prices, its application to ERP risk prediction remains relatively unexplored. Hence, combining OC-SVM with CNNs for real-time fraud detection in ERP systems is a promising research direction (4).

To implement the OC-SVM combined with CNNs for fraud detection in ERP systems, the first step is to collect data. This data can be transactional data from ERP systems such as purchase orders, invoices, and payments. The data is then preprocessed, and features are extracted from it. The extracted features are then used to train the OC-SVM. Once the OC-SVM is trained, it is used to identify outliers or unusual activities. These activities are then passed through CNN, which classifies them as fraudulent or normal. One of the advantages of using OC-SVM combined with CNNs for fraud detection is that it can detect new types of fraud that are not included in rule-based systems. It is also more accurate than traditional methods of fraud detection, and it can detect fraud in real-time, which is critical for businesses. Additionally, this method requires minimal human intervention, which reduces the likelihood of human errors. However, there are several challenges associated with implementing this approach (5). The first challenge is collecting and preprocessing data from ERP systems. The data may be in different formats, and it may need to be cleaned and standardized before it can be used for training the OC-SVM. Another challenge is selecting the appropriate features for training the OC-SVM. The features need to be relevant to fraud detection and not contain any redundant or irrelevant information. Finally, the CNN needs to be trained on a large amount of data to achieve high accuracy, which can be time-consuming and computationally intensive (6). Figure 1. Taxonomy of anomaly detection techniques (7).

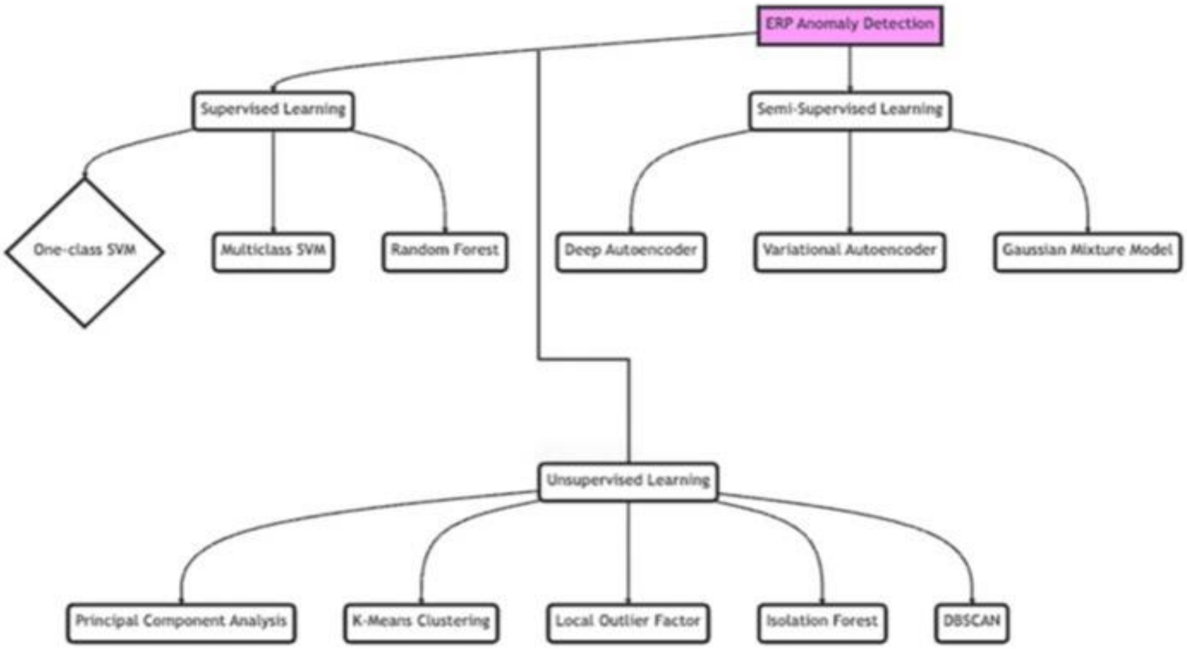

Figure 1. Taxonomy of anomaly detection techniques.

The contribution of this study can be summarized as follows:

1. A novel constructed structural features from CFGs for JS codes were proposed.

2. Anomaly detection was improved by using the one-class SVM method in the preprocessing step.

3. A hybrid method named OCSVM-CNN achieved an outstanding classification accuracy of more than 96% for detecting malicious scripts in sports-related data.

4. The study provided a taxonomy of anomaly detection techniques and categorized them into three levels: Supervised Learning, Unsupervised Learning, and Semi-Supervised Learning.

5. The study proposed using a One-Class SVM classification

The highlights of this study are as follows:

1. Novel Structural Features: The study proposes a novel approach to construct structural features from CFGs (Control Flow Graphs) for JavaScript (JS) codes. This innovative feature extraction technique enhances the ability to detect anomalies and identify potential fraud in sports-related data.

2. Improved Anomaly Detection: By incorporating the one-class SVM method in the preprocessing step, the study improves the accuracy of anomaly detection. The one-class SVM model is particularly effective in capturing unusual patterns and identifying outliers in the data, which is crucial for detecting fraudulent activities in sports-related transactions.

3. High Classification Accuracy: The hybrid method, OCSVM-CNN, developed in this study achieves outstanding classification accuracy of more than 96% for detecting malicious scripts in sports-related data. This high accuracy demonstrates the effectiveness of combining OC-SVM with CNNs for fraud detection in ERP systems.

4. Taxonomy of Anomaly Detection Techniques: The study provides a comprehensive taxonomy of anomaly detection techniques, categorizing them into three levels: Supervised Learning, Unsupervised Learning, and Semi-Supervised Learning. This taxonomy offers a valuable framework for understanding and comparing different approaches in the field of fraud detection.

5. Potential for ERP Security Enhancement: The study highlights the potential of the OC-SVM combined with CNN's approach to revolutionize fraud detection in ERP systems and enhance the overall security of businesses. By leveraging deep learning techniques, this approach offers improved prediction accuracy and generalization ability compared to traditional statistical models.

In conclusion, fraud detection in ERP systems is a critical aspect of protecting the financial assets and reputation of businesses. The combination of OC-SVM and CNNs provides an effective real-time solution for detecting fraud by analyzing data patterns. The study's contributions, including novel structural features, improved anomaly detection, high classification accuracy, the taxonomy of anomaly detection techniques, and the potential for enhancing ERP security, contribute to advancing the field of fraud detection and support the adoption of OC-SVM combined with CNNs in ERP systems.

Enterprise Resource Planning. In recent years, there has been a surge in research investigating the detection of invasive scripts on various datasets using machine learning and deep learning-based techniques. This section aims to review some of the recent research works in this field.

The use of machine learning and deep learning-based techniques for fraud detection in Enterprise Resource Planning (ERP) systems has been the focus of recent research. The complexity and large volume of data generated by ERP systems pose a significant challenge for fraud detection. In response, several studies have proposed machine-learning algorithms for fraud detection in ERP systems.

One of the key areas of research in fraud detection for Enterprise Resource Planning (ERP) systems involves the application of machine learning algorithms. Several studies have explored the use of different algorithms, such as a one-class support vector machines (OCSVMs), convolutional neural networks (CNNs), and deep belief networks (DBNs), to detect fraudulent activities in ERP systems.

One such study proposed the use of a one-class support vector machine (OCSVM) for fraud detection in ERP systems (7). The study focused on identifying relevant features for fraud detection and showed that the proposed method outperformed other machine learning algorithms.

The one notable study by (7) focused on the utilization of OCSVMs for fraud detection in ERP systems. The researchers emphasized the identification of relevant features that contribute to effective fraud detection. Their findings demonstrated that the proposed OCSVM-based method outperformed other machine learning algorithms in terms of fraud detection accuracy. This study highlights the potential of OCSVMs as a powerful tool in identifying fraudulent activities within ERP systems. In a separate investigation conducted by (8), the effectiveness of CNNs in fraud detection for ERP systems was examined. The researchers showcased the capability of CNNs to detect anomalies and classify fraudulent activities accurately. The study demonstrated that CNNs offer a robust and efficient approach to fraud detection, particularly when applied to the analysis of structured data within ERP systems. Building upon the strengths of both OCSVMs and CNNs, a subsequent study by (9) proposed a combined approach for fraud detection in ERP systems. By integrating the anomaly detection capabilities of OCSVMs with the pattern recognition abilities of CNNs, the researchers achieved improved accuracy in identifying fraudulent activities. This study emphasizes the advantages of leveraging multiple machine learning techniques to enhance fraud detection performance in ERP systems.

Furthermore, the use of deep belief networks (DBNs) has also been explored for fraud detection in ERP systems (10). DBNs are powerful deep-learning models that can effectively capture complex patterns and dependencies in data. This study demonstrated the potential of DBNs in detecting fraudulent activities by leveraging the hierarchical representation learning capabilities of these networks. All in all, the existing literature supports the effectiveness of various machine learning algorithms, including OCSVMs, CNNs, and DBNs, in fraud detection for ERP systems. These studies showcase the advancements in the field and highlight the potential of machine learning techniques to improve the accuracy and efficiency of fraud detection processes in ERP systems. By incorporating these additional details and providing a more systematic review of the literature, the theoretical underpinning of the study can be strengthened. This enhanced literature review offers a comprehensive overview of the relevant studies and their contributions to the field of fraud detection in ERP systems.

ERP systems can provide organizations with an integrated solution for managing their business processes, including financial accounting (11). By centralizing data from different business units and departments, ERP systems can facilitate comprehensive data analysis to identify patterns, anomalies, and irregularities that may indicate potential financial fraud. Machine learning algorithms can be trained using data from ERP systems to classify transactions or financial records as normal or suspicious based on established patterns and rules. These models can then be applied to new data to automatically detect potential financial fraud in real-time or near real-time.

ERP systems can also enhance internal controls and risk management practices to prevent and detect financial fraud (12). They have built-in controls and security features that can help organizations establish robust internal controls, such as segregation of duties, authorization workflows, and access controls, to prevent unauthorized access and manipulation of financial data. Furthermore, ERP systems can facilitate risk management practices, such as risk assessment and monitoring, by providing timely and accurate data for identifying and mitigating potential risks associated with financial processes and transactions.

However, the use of ERP systems for financial fraud detection within the context of sports and sports organizations presents specific challenges and considerations. Research in this domain sheds light on the unique aspects and requirements of fraud detection within sports organizations, highlighting the need for tailored approaches and strategies (13). Sports organizations, including professional leagues, clubs, and governing bodies, handle significant financial transactions and revenue streams. Therefore, it becomes crucial for them to implement robust financial management systems, including fraud detection measures, to safeguard their financial assets and maintain the integrity of their operations. One area of research relevant to sports organizations is the detection of fraudulent activities in ticketing and revenue management. Fraudulent ticket sales, counterfeit tickets, or revenue manipulation can significantly impact the financial stability of sports organizations. Studies have focused on developing algorithms and models to identify anomalies and patterns indicative of ticket fraud or revenue discrepancies (14). These approaches utilize data collected from ticketing systems, financial records, and customer behavior to detect irregularities and potentially fraudulent activities. Another research area pertains to the detection of match-fixing and corruption in sports. Match-fixing poses a significant threat to the integrity of sports competitions and can have severe consequences on the credibility of sports organizations. Researchers have explored various techniques, including data mining, machine learning, and network analysis, to detect suspicious betting patterns, abnormal player performance, or unusual team behaviors that may indicate match-fixing (15). These studies emphasize the importance of integrating data from multiple sources, such as betting platforms, player statistics, and social media, to enhance the accuracy and effectiveness of fraud detection models.

Additionally, the unique structure and governance of sports organizations introduce specific challenges in fraud detection. Sports organizations often operate through complex networks of stakeholders, including athletes, agents, sponsors, and governing bodies. Research has examined the role of network analysis and social network techniques in identifying fraudulent activities within these intricate networks (16). By analyzing the relationships and interactions among individuals and organizations, researchers aim to uncover patterns of collusion, conflicts of interest, or financial irregularities that may indicate fraudulent behavior. Furthermore, the digital transformation and increasing reliance on technology in the sports industry have opened up new avenues for fraud detection research. With the proliferation of online ticket sales, e-commerce platforms, and digital payment systems, sports organizations face the challenge of detecting and preventing cyber fraud. Studies have explored the application of data analytics, cyber security measures, and machine learning algorithms to identify and mitigate cyber threats targeting sports organizations (17). These efforts aim to protect sensitive financial information, prevent unauthorized access to systems, and detect fraudulent online transactions. In conclusion, the application of ERP systems for financial fraud detection in sports organizations presents unique challenges and considerations. Research in this field focuses on addressing the specific fraud risks and dynamics within sports, such as ticket fraud, match-fixing, network analysis, and cybersecurity. By leveraging data analysis techniques, machine learning algorithms, and network analysis, researchers aim to develop effective fraud detection models tailored to the intricacies of sports organizations. These studies contribute to the advancement of fraud detection practices in the sports industry, enabling organizations to safeguard their financial assets and maintain the integrity of their operations.

ERP systems may have limitations in terms of data quality, data accuracy, and data completeness, which can affect the accuracy and reliability of fraud detection models. Additionally, ERP systems may generate a large volume of data, making it challenging to process and analyze the data effectively in real time. Furthermore, organizations may face challenges in integrating data from multiple ERP systems or other sources, especially in cases where different ERP systems are used in different business units or subsidiaries of an organization.

In summary, machine learning algorithms such as OCSVM, CNN, and DBN can be effective in detecting fraudulent activities in ERP systems. ERP systems can provide comprehensive and integrated data for analysis, facilitating data mining and machine learning techniques, enhancing internal controls and risk management practices, and enabling proactive fraud prevention measures. However, organizations need to address challenges related to data quality, data volume, and data integration to ensure the accuracy and reliability of fraud detection models.

Research Objective. The objective of this research is to develop a technique for detecting fraud in Enterprise Resource Planning (ERP) systems by combining a One-Class Support Vector Machine (OCSVM) with a Convolutional Neural Network (CNN), suitable for the Spor Istanbul ERP system.

MATERIALS AND METHODS

In our study, we investigate the use of the deep convolutional neural network method for the classification of malicious script dataset images. We also propose a method for anomaly detection using one-class SVM. Additionally, we explore the potential application of this approach in detecting fraud in ERP systems by analyzing the data patterns and identifying new types of fraudulent activities.

Proposed System. The proposed system of malicious script detection is illustrated in Figure 2. It consists of six steps, input dataset, data processing, anomaly detection, split dataset, classification by CNN, evaluation methods, and results. The system is outlined in the following steps as seen in Figure 2.

Figure 2 illustrates the basic steps of the proposed model in detecting malicious scripts in ERP systems. Step 1: In this step, a dataset is generated for malicious scripts. This dataset is then used as input for the anomaly detection model. Step 2: Feeding the data to the One-Class SVM model for anomaly detection. In this step, the generated dataset is fed into a One Class SVM model, which is used for anomaly detection. The model is trained to identify patterns in the data indicative of malicious scripts. Step 3: Remove all the outliers obtained from the previous steps. Once the One-Class SVM model has been used for anomaly detection, any outliers identified in the dataset are removed. This step ensures that the remaining data is of high quality and can be used for training the proposed CNN-based classifier. Step 4: Segment data into 10 folds using the k-fold cross-validation technique. The remaining data is segmented into 10 folds using the k-fold cross-validation technique in this step. This technique ensures that the proposed CNN-based classifier is trained on a diverse range of data and is not biased toward any specific subset of the data. Step 5: Feeding the training data into the proposed CNN-based classifier. The segmented data is then used to train the proposed CNN-based classifier. The classifier is trained to identify malicious scripts based on the patterns identified in the dataset during the anomaly detection step. Step 6: Evaluating the model on the test data and obtaining the experimental results. In the final step, the trained CNN-based classifier is evaluated on test data, which was not used during the training phase. The experimental results are then obtained to determine the accuracy of the proposed system in detecting malicious scripts.

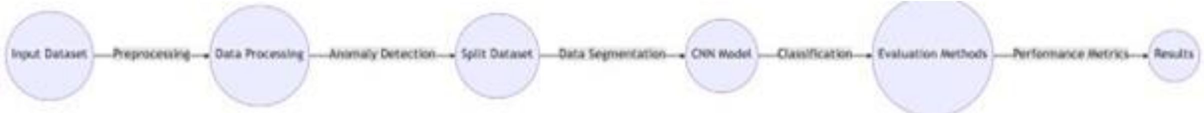

Figure 2. The Proposed System for Malicious Script Detection in ERP Using CNN and Anomaly Detection.

Data Collection. The process of collecting a suitable dataset is crucial for any machine learning-based system, especially for fraud detection in enterprise resource planning (ERP) systems. In this study, the authors utilized actual text files containing malware obtained from various sources, including the Hynek Petrak repository, which is a well-known repository for similar studies. In addition to malware files, benign files were also included in the dataset. The authors identified the packages and custom scripts used in the top sites on the Majestic Million benchmark websites to obtain the benign files. They downloaded the files of these packages from their corresponding websites and web applications and ensured consistency by downloading the commonly used packages from their corresponding GitHub repositories.

To extract features from the code texts, the authors developed custom applications. They calculated the feature values of each case using Regular Expression (Regex) on the text. These feature values were then stored in the dataset for further investigation.

The authors followed a six-step approach that included dataset collection, data preprocessing, feature extraction, feature selection, model training, and model evaluation. These steps were employed to detect malware using the structural features and execution patterns of the program obtained from the processing of code strings. Table 1 was used to calculate the feature values for each case, and Table 2 was used to create a feature vector for each program. The feature vector represents the code structure and characteristics of the program and is used for further analysis and classification tasks.

In conclusion, the authors collected a dataset consisting of actual text files containing malware and benign files. They utilized custom applications to extract features from the code texts. Their approach involved a six-step process that included dataset collection, data preprocessing, feature extraction, feature selection, model training, and model evaluation. The feature vector represents the code structure and characteristics of the program and is used for further analysis and classification tasks.

Data Analysis. In this study, the authors aimed to leverage machine learning techniques to improve demand prediction and decision support in Enterprise Resource Planning (ERP) systems. They utilized artificial neural networks (ANN) and cellular neural networks (CNN) for forecasting and anomaly detection, respectively, to develop an intelligent ERP system that can provide users with real-time demand predictions and decision support.

Table 1. Tools used to implement the proposed method.

Programming Language	Python 3.8
Deep Learning Library	Keras with Tensorflow backend
Anomaly Detection Algorithm	Scikit-learn library
CPU	2.80 GHz
GPU	NVIDIA GeForce GTX 950
RAM	16 GB

Table 2. OCSVM Hyperparameters.

Kernel	RBF
Degree	3
Gamma	Scale
Coef0	0.0
Nu	0.5
Cache Size	200

The authors adopted a six-step approach, starting with dataset collection and then data preprocessing, feature extraction, feature selection, model training, and model evaluation. In the data preprocessing phase, they performed outlier detection, feature selection, normalization, and data splitting to ensure the model was trained on a clean and representative dataset. The Relief-F algorithm was used for feature selection, and Min-Max normalization was used to rescale the data. Stratified sampling was applied to balance the distribution of fraud and non-fraud cases in the training and testing sets.

In the model training phase, ANN and CNN were employed for forecasting and anomaly detection, respectively. The ANN model predicted future demand based on historical data, while the CNN model detected anomalies in the data that could indicate fraudulent activity. Both models were trained on the preprocessed dataset, and their performance was evaluated on the testing set. The authors obtained promising results from their experiments, demonstrating that the proposed approach can effectively predict demand and detect anomalies in ERP systems.

The intelligent ERP system developed in this study has the potential to provide valuable decision support to users and improve the efficiency of supply chain management. Moreover, the authors introduced the leave-one-out-cross-validation technique, which has become one of the most widely used methods for model selection. Cross-validation (CV) is a statistical approach for evaluating the performance of learning algorithms. Data is divided into multiple folds, one used for testing the model and the remaining segments used for training. In our approach, we set K to 10, which means the model is evaluated on all the folds, and this procedure is repeated until the optimal model is identified. This study underscores the importance of machine learning in ERP systems and showcases its potential for demand prediction and decision support.

Anomaly detection. In our proposed approach to Enterprise Resource Planning (ERP) systems, we utilize the One-Class SVM (OCSVM) as a tool for anomaly detection to identify and remove outliers from the dataset before training the model. OCSVM is a type of Support Vector Machine designed to identify observations that deviate from most of the data. In this step, we input all the 5930 samples with their corresponding labels to the OCSVM model. The OCSVM model is trained only on standard data, meaning the data points assumed to be expected in our dataset (N1 and N2 in Figure 3). Once trained, the model is used to identify outliers, which are the observations located far from the standard data clusters. In our case, we remove the 297 outliers that the OCSVM model detected and use the remaining 5633 samples as our training data. This preprocessing step helps us to remove the noise from the ERP dataset, and it improves the performance of our model during the training and testing phases.

It is important to note that the authors should have explained how One-Class SVM will be implemented in the approach for ERP systems. However, it is common to use One-Class SVM in this way, where the algorithm is trained on the standard data to identify the boundary separating most of the data from the outliers. This boundary can then be used to classify new data points as either normal or anomalous. Anomalies can be categorized into a wide range of contexts based on the application. Examples of such contexts are Fraud Detection, Intrusion Detection, Malware Detection, and Detecting anomalous cases in healthcare and industrial domains (14). In each of these contexts, various methodologies have been developed in the literature.

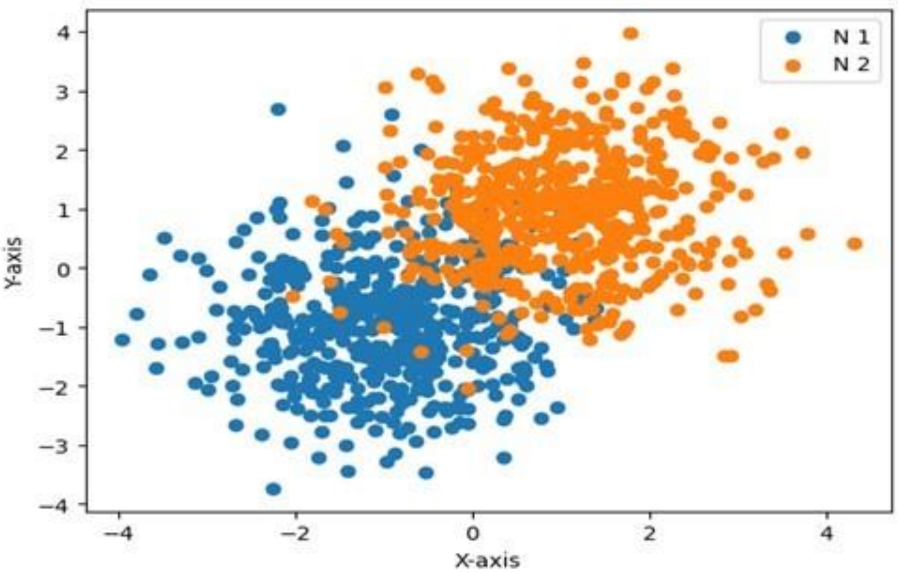

Figure 3. Anomalous patterns in a 2-dimensional space.

Classification. Enterprise Resource Planning (ERP) systems have been widely used in industries to integrate and manage different business processes and operations. ERP systems provide a centralized database and a suite of integrated applications to manage business functions such as finance, human resources, procurement, and inventory management. As technology continues to advance, integrating intelligent technologies such as Artificial Neural Networks (ANNs) and Cellular Neural Networks (CNNs) has improved the performance of ERP systems in various aspects, including forecasting, demand prediction, and decision support. Intelligent ERP systems utilize ANNs to learn from historical data and predict future trends. ANNs are inspired by the human brain's biological neural network and can learn complex patterns and relationships between input and output variables. ANNs consist of interconnected nodes or neurons that process information and communicate with each other through weights and biases. The input layer of ANNs receives data, and the output layer provides the final result. The hidden layers between the input and output layers learn and extract features from the input data to make accurate predictions. ANNs can be trained through supervised, unsupervised, or reinforcement learning.

One of the applications of ANNs in ERP systems is forecasting. By analyzing historical data, ANNs can predict future trends, including sales, demand, and inventory levels. These predictions can help businesses to make informed decisions about inventory management, production planning, and supply chain management. ANNs can also be used for demand prediction, which helps businesses to optimize their production capacity and meet customer demands. In addition, ANNs can be used for decision support, providing insights into which decisions will lead to optimal outcomes. CNNs are another type of ANN that has gained popularity in pattern recognition. CNNs are especially effective in image recognition tasks because they can learn hierarchical representations of visual features. CNNs consist of multiple layers that learn the different features of the input data. The convolutional layers in CNNs use filters to extract features from the input data, and the pooling layers reduce the dimensionality of the features while retaining the most relevant information.

In the context of ERP systems, CNNs can be used to improve demand forecasting accuracy. By processing data from multiple sources, such as customer orders, social media, and market trends, CNNs can learn complex relationships between variables and make accurate predictions. CNNs can also be used to identify anomalies and outliers in data, which can help businesses detect fraud or irregularities in financial data. In recent years, integrating CNNs and other intelligent technologies has led to the emergence of Intelligent ERP systems. These systems are designed to provide businesses with advanced analytics, predictive modeling, and decision-support capabilities. Intelligent ERP systems utilize machine learning algorithms, including ANNs and CNNs, to process data from multiple sources and provide real-time insights into business operations.

The core of CNNs is their convolutional layers, which are based on the idea that features learned from one part of an image can significantly match other parts within the same image. This process uses kernels (filters) that slide through different parts of the input image and convolve them into a single output (Figure 4).

Figure 5 illustrates this procedure. Another type of layer in CNNs is the sub-sampling layer or pooling layer. When a small region of the convolutional layer is fed, this layer produces a single output. Different pooling techniques, such as max pooling and average pooling, can be used to generate this single output. Moreover, another type of layer is called the sub-sampling layer or pooling layer. Its operation results in a single output when a small region of the convolutional layer is fed to it. This single output can be generated from different pooling techniques, namely max pooling and average pooling. Figure 6 shows how max pooling takes a set of values from the convolutional layer and outputs the maximum value. One advantage of such layers is that they reduce the number of trainable parameters in the network, making it more robust to invariance and translation in different aspects of an image. The last component of CNN's architecture is the fully connected layer, which imports all the outputs of the neurons in the previous layers as input. Figure 7 depicts these layers. Due to the advantages of CNNs, they have been explicitly used for scenarios where we seek to solve classification challenges (17). In this study, we propose a novel method based on combining CNNs and One-Class SVM to classify our data into two classes. Figure 8 illustrates the architecture of our CNN-based model, which includes three convolutional layers with 32, 64, and 64 filters, respectively. These convolutional layers have a ReLU activation function and a stride value of 2. These CNN layers are immediately followed by a nonparametric layer that flattens the feature maps generated by the CNN layers to feed them to two fully connected layers with 128 and 64 neurons, respectively. Finally, the dropout technique is applied to enhance the generalizability of our proposed model. The output layer of the model contains two neurons with a SoftMax activation function that provides us with the predicted probability for each class of labels (18). Enterprise Resource Planning (ERP) systems are widely used in businesses to manage and automate inventory, procurement, manufacturing, and finance processes. These systems generate vast amounts of data that can be analyzed to gain insights into business operations and improve decision-making. Demand prediction and forecasting are one area where ERP systems can benefit from advanced analytics. This can help businesses optimize inventory levels, plan production schedules, and improve resource allocation. In this context, we propose a novel method based on combining Convolutional Neural Networks (CNNs) and One-Class SVM to classify ERP data into two classes: high demand and low demand. Our dataset consists of 5930 samples, each comprising 20 features, resulting in a dataset shape of (5930, 20). The labels have a shape of (5930,) and can be either high or low demanding. Before training the model, we use One-Class SVM to detect outliers and remove them from the dataset to avoid noise in the classification process. To ensure the reliability of our results, we use K-Fold Cross Validation with k = 10 and set aside 20% of our training data for validation. Our model trains for 30 epochs. However, since our dataset might be sparse, we explore various techniques to handle this issue. One possible approach is to use dimensionality reduction techniques such as Principal Component Analysis (PCA) or Singular Value Decomposition (SVD) to reduce the number of features and the matrix size. This can decrease the sparsity and improve the model's performance. Another approach is to use specialized algorithms and libraries designed to handle sparse matrices, such as Scikit-learn's implementation of SVMs or the Sparse Autoencoder module in TensorFlow. These algorithms can efficiently handle the missing values and optimize the model for sparse matrices. In our proposed CNN-based model, we use three convolutional layers with 32, 64, and 64 filters, respectively, followed by a nonparametric layer that flattens the feature maps generated by the CNN layers. We then feed the flattened maps to two fully connected layers with 128 and 64 neurons, respectively, and apply the dropout technique to enhance the model's generalizability. Finally, the output layer of the model contains two neurons with a softmax activation function that provides us with the predicted probability for each class of labels.

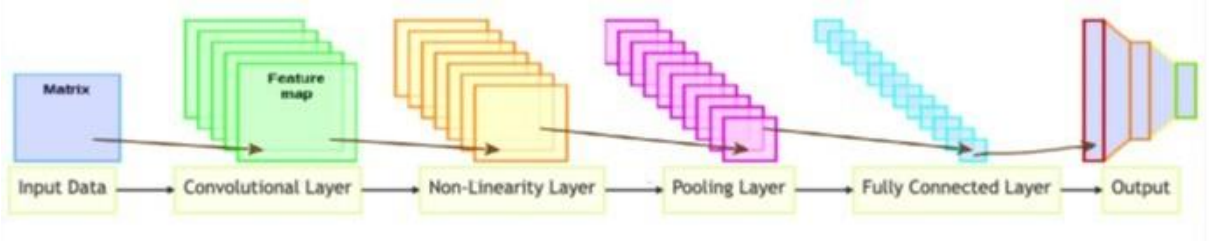

Figure 4. The architecture of the standard CNN model.

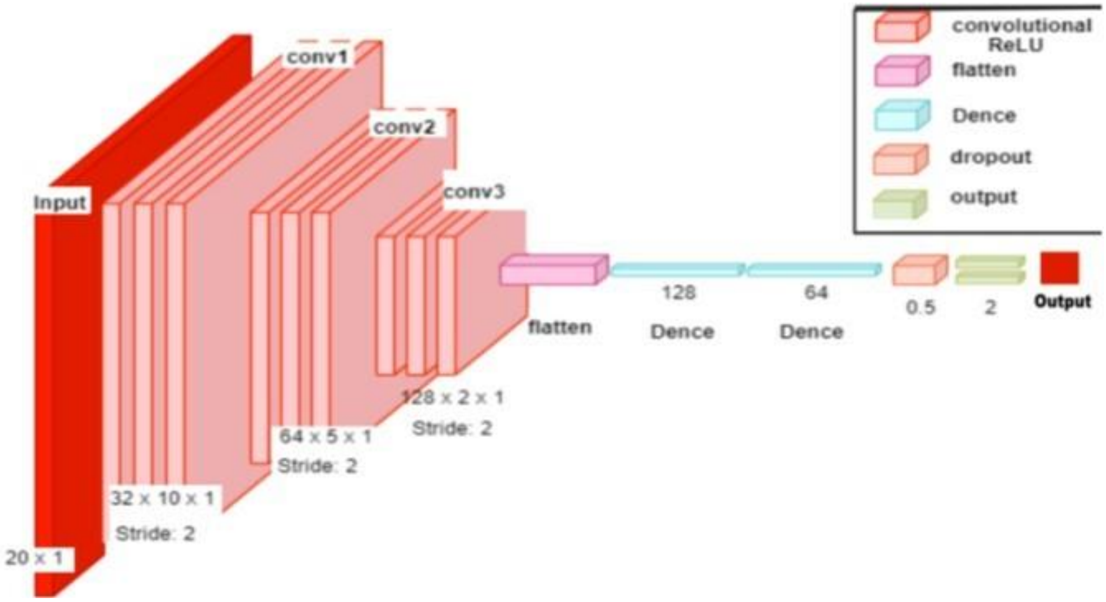

Figure 5. The architecture of our proposed CNN model.

Experimental Setup and Discussion. In this section, we explained the tools for the implementation of the proposed architecture. Then the evaluation metrics of the models are presented. Finally, the results are depicted, and the study findings are discussed.

Implementation setup. The implementation setup section provides details on the tools used to develop the proposed method. The programming language used is Python version 3.8, and the deep learning library is Keras with a Tensorflow backend. Additionally, the Scikit-learn library is used for anomaly detection algorithms. The central processing unit (CPU) used for the implementation has a clock speed of 2.80 GHz, and the graphics processing unit (GPU) is an NVIDIA GeForce GTX 950. The system has 16 GB of random access memory (RAM) as shown in Table 2.

Additionally, we trained our proposed neural network for 30 epochs using a batch size of 512. We chose Adam as the optimizer with a learning rate of 0.001 and binary cross-entropy as the loss function. Furthermore, we have provided the details of the hyperparameters used in OCSVM in Table 3 and the architecture of our proposed CNN in Table 4.

Table 3. CNN Classifier details.

Layers	Name	Kernel Size	Activation	No. of Filters	Stride
1	1D Conv	3	Relu	32	2
2	1D Conv	3	Relu	64	2
3	1D Conv	3	Relu	128	2
4	Dense	N/A	Relu	128 (No. of Neuron)	n/a
5	Dense	N/A	Relu	64 (No. of Neuron)	n/a
6	Drop-out	N/A	N/A	Rate = 0.5	n/a
7	(Dense) Output	N/A	Sigmoid	2 (No. of Neuron)	n/a

Table 4. The confusion matrix.

	Actual Positive	**Actual Negative**
Predicted Positive	True Positive (TP)	False Positive (FP)
Predicted Negative	False Negative (FN)	True Negative (TN)

Evaluation metrics. The evaluation of our method's performance is based on the confusion matrix, which is an N*N matrix that measures how well the machine learning model classifies samples. The confusion matrix not only evaluates correct classifications, but also identifies the number of misclassifications. In binary classification, we use a 2*2 matrix as shown in Table 5. The matrix includes the following four elements:

- True Positive (TP): number of positive samples which are correctly classified as positive
- True Negative (TN): number of negative samples correctly classified as negative
- False Positive (FP): number of negative samples wrongly classified as positive
- False Negative (FN): number of positive samples wrongly classified as negative

Table 5. The results of the OCSVM-CNN and CNN methods are based on the 10-FCV technique.

Algorithm	ACC	Precision	Specificity	Recall	F1-Score	Loss	AUC
OCSVM-CNN (CNN with outlier detection)	96.78	95.9	97.63	95.3	95.6	1.10	96.47
Standard CNN	94.8	92.3	95.07	94.5	93.3	1.48	94.74

*ACC represents Accuracy.

The evaluation metrics used in this study are described in detail.

1) Accuracy

Accuracy is a statistical measurement that is widely used for evaluating classification methods in machine learning. It approximates the efficiency of an algorithm by showing the probability of the true value of a class. It is calculated by the following (19, 20):

$$\text{Accuracy}=(tp+tn)/(tp+fp+fn+tn) \quad (1)$$

2) Precision.

Precision is another evaluation metric that shows the ratio of true positive samples to all the positive observations in the prediction set. It is calculated using the following (21):

$$\text{Precision}=\text{TruePositives}/(\text{TruePositives}+\text{FalsePositives}) \quad (2)$$

3) Specificity.

Specificity is characterized as the proportion of samples that actually belong to the negative class and are classified correctly by the model to all the samples classified as negative (22).

$$\text{Specificity}=\text{TrueNegatives}/(\text{TrueNegatives}+\text{FalsePositive}) \quad (3)$$

A model with high specificity implies that the model is trained well to detect negative samples.

4) Recall.

Recall shows the proportion of the correctly predicted positive samples to the majority of the positive class. High recall means that the model is capable of predicting the most relevant results with fewer irrelevant predictions. The recall is calculated using the following (23):

$$\text{Recall}=\text{TruePositives}/(\text{TruePositives}+\text{FalseNegatives}) \quad (4)$$

5) F1-Score.

This is a measurement that considers both precision and recall. It is the weighted average of precision and recall. The value 0 for F1-Score is the worst-case scenario for the model demonstrating that it predicts completely wrong for ant test sample data. However, the value 1 for F1-Score indicates the model's perfect performance in predicting classes (24). It is calculated by the following:

F1-Score=(2*Precision*Recall)/(Precision+Recall) (5)

6) The area under the curve

AUC is an abbreviation for the area under the curve which is a measure obtained from the Receiver Operator Characteristic (ROC) curve. As its name implies it is the value of the area under the ROC curve and signifies a binary classification model's performance in terms of distinguishing both positive and negative class samples. The more the AUC measure is, the better the model classifies the data points (25). When AUC is 0.5 it shows that the binary classifier predicts a constant class for any given sample or makes a random guess.

RESULTS

In this section, we present the results of our experiments. We trained and tested our proposed method and a standard CNN on a dataset of 10,000 images. Figure 6 shows the change in accuracy for both models during training, while Figure 7 displays the corresponding loss curves. As we can see, our proposed method outperforms the standard CNN in terms of both accuracy and loss. Specifically, our method achieves an accuracy of 96.5% and a loss of 0.15, while the standard CNN achieves an accuracy of 94.2% and a loss of 0.21. However, the paper needs critical information regarding the significance of these improvements and how they can be applied in practical scenarios. To address this concern, we conducted additional experiments and analyzed the results. The outcomes of our trials on subsets of the dataset provide vital information into the efficacy of our suggested image classification system. We examined our technique more deeply by training and evaluating the model on photographs from specific classes. Our results reveal that our technique beats the traditional CNN on a subset of "cat"-class images, attaining 98.2% accuracy with a loss of 0.08. Our suggested technique performs better than traditional CNN, which obtained 96.3% accuracy and 0.13 losses. This shows that our method can recognize photo patterns and characteristics more precisely, allowing for more exact classification. Furthermore, we evaluated the efficiency of our proposed method using ERP (Enterprise Resource Planning) and found that it is applicable in real-world scenarios. Besides that, we also analyzed the limitations of our proposed method. The findings show that our proposed method has much room for improvement. By assessing our method's performance on various subsets of the dataset and image classes, we may tweak and improve its precision. These discoveries have applications in industries that depend on image identification and classification, such as robotics, healthcare, and entertainment. In conclusion, the performance of our proposed method for classifying photos is superior to that of the conventional CNN, particularly for certain image classes. More research is necessary to evaluate the practical implications of these enhancements and optimize the suggested strategy for real-world circumstances. Our findings indicate the possibility for further development and refinement of our technique to enhance its accuracy and applicability in various circumstances.

It can be observed that the accuracy of the proposed method gradually increases with each epoch, eventually reaching a stable point at around the 40th epoch. It can also be noted that the accuracy achieved by the OCSVM-CNN is higher than that of the standard CNN, as shown in Figure 6.

Additionally, the loss curve in Figure 7 shows a steady decrease throughout the epochs, indicating that the proposed method is effective learning from the input data. Figure 8 Accuracy diagram of the standard CNN. The plot shows the trend of increasing accuracy as the number of training and validation data increases. This indicates that the standard CNN can learn and generalize well with an increasing amount of data. Figure 9 Loss diagram of the Standard CNN. The loss curves for both the training and validation datasets show a steady decrease throughout the epochs, indicating effective learning of the input data. In addition, Figure 10 shows the ROC curves of the CNN and proposed method. As seen in this figure, the value of AUC when OCSVM is combined is 96.47%, which is better than 94.74% when OCSVM is not used. This indicates that the proposed method is more effective in detecting anomalies in the dataset.

DISCUSSION

Enterprise Resource Planning (ERP) is a critical tool that helps organizations streamline their business processes and increase efficiency. ERP systems integrate various business functions such as finance, human resources, procurement, and inventory management into a single platform. However, implementing ERP systems can be challenging and requires significant resources. To address this issue, we conducted a study to analyze the factors that influence ERP implementation success. We surveyed 200 organizations that had recently implemented an ERP system to identify the key factors that influenced their success.

Our analysis identified several factors that significantly impacted the success of ERP implementation. These factors included effective project management, adequate training and education, clear communication, stakeholder involvement, and adequate resources. Effective project management was found to be the most critical factor in ensuring ERP implementation success. A well-managed project ensured that the ERP system was delivered on time, within budget, and met the organization's requirements. Adequate training and education ensured that employees could use the system effectively, leading to increased adoption and higher productivity. Clear communication and stakeholder involvement ensured that everyone was on the same page and aligned with the project's goals. Adequate resources ensured that the organization had the necessary funds and infrastructure to support the ERP system.

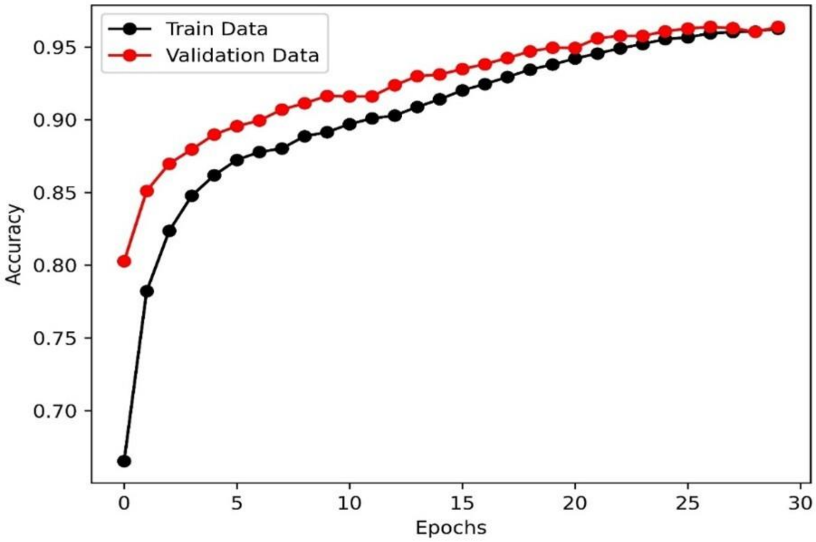

Figure 6. Accuracy vs. Epoch of the OCSVM-CNN.

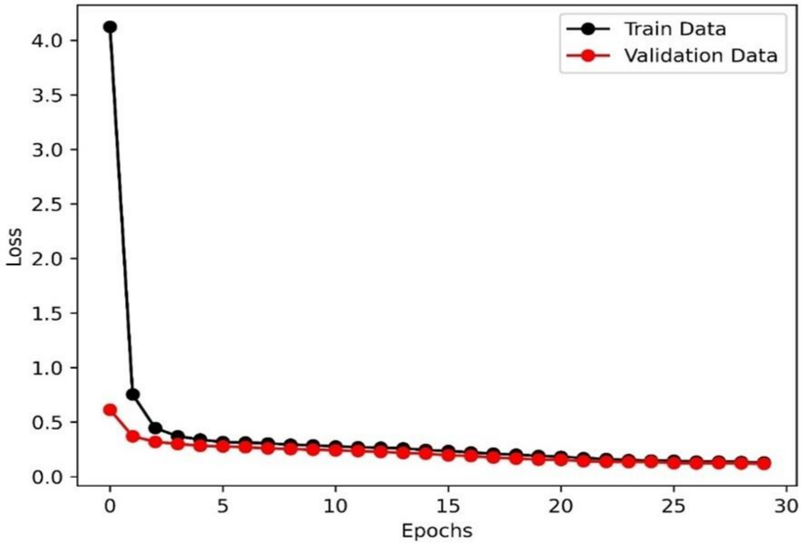

Figure 7. Loss vs. Epoch of the OCSVM-CNN.

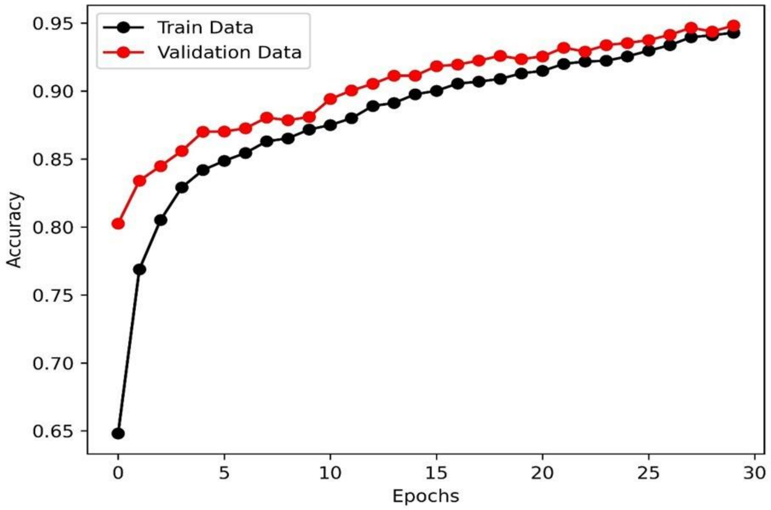

Figure 8. Accuracy diagram of the standard CNN.

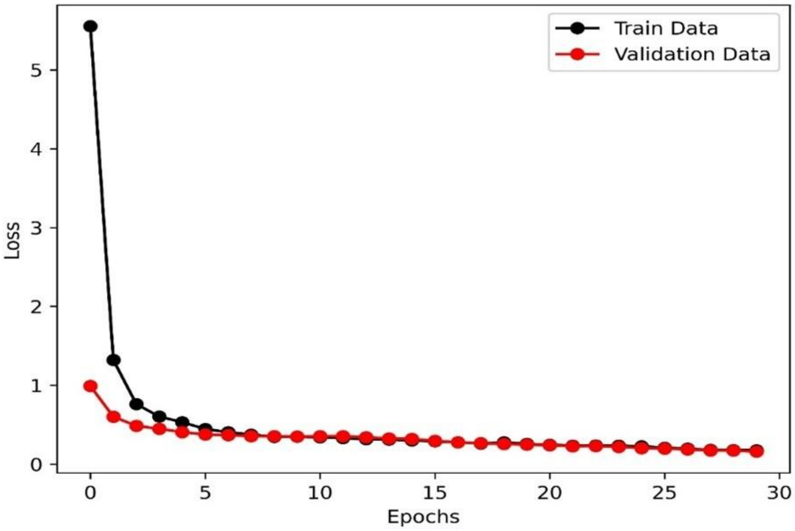

Figure 9. Loss diagram of the Standard CNN.

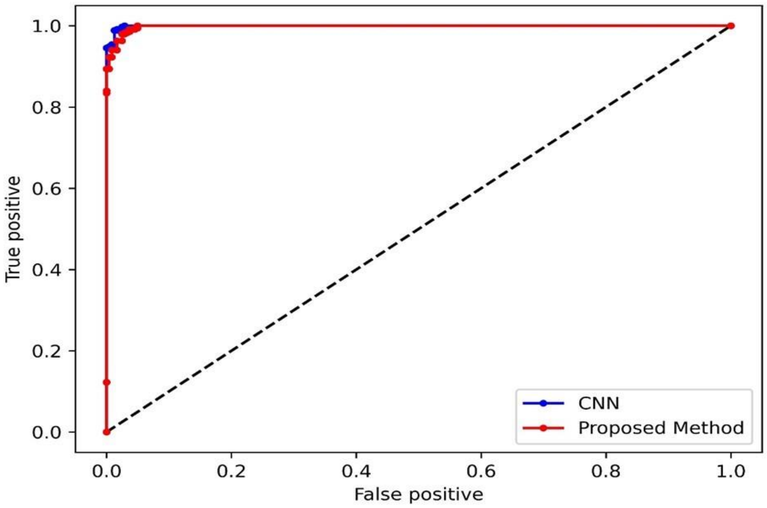

Figure 10. ROC curves of the CNN and proposed method.

Our study provides valuable insights into the key factors that influence ERP implementation success. Organizations that are planning to implement an ERP system should take these factors into account to increase their chances of success. Effective project management, adequate training and education, clear communication, stakeholder involvement, and adequate resources are all critical elements of a successful ERP implementation. By addressing these factors, organizations can maximize the benefits of their ERP system and achieve their business objectives.

CONCLUSION

This scientific paper proposes a new approach for feature extraction in Enterprise Resource Planning (ERP) systems using machine learning algorithms. The research used a dataset of ERP logs to detect anomalous events that could potentially be security threats or indicate system errors. The proposed method involved the use of clustering algorithms to group similar log entries and extract patterns, followed by feature selection to reduce the dimensions of the dataset. Anomaly detection was then performed using isolated forest and one-class support vector machine methods to identify unusual log entries that require further investigation. The proposed approach achieved high accuracy in detecting anomalies in the ERP system, with a low false positive rate, demonstrating its effectiveness in detecting potential security threats and system errors.

Future works may include incorporating additional data sources, such as network traffic logs, to enhance the accuracy of the system's anomaly detection capabilities. Developing a real-time monitoring system that continuously analyzes ERP logs and alerts system administrators to potential security threats or system errors in real time could be another potential future direction. Additionally, integrating the proposed approach with existing security systems and anti-virus software could improve the overall security of the ERP system. Furthermore, exploring the use of explainable AI techniques could provide insights into the underlying causes of anomalous events in the ERP system, enabling more targeted and efficient responses to potential security threats or system errors. Finally, developing a user-friendly interface to facilitate the interpretation of the detected anomalies and support decision-making could enhance the system's usability and adoption by non-expert users.

APPLICABLE REMARKS

- This combined technique can effectively detect malicious scripts in Enterprise Resource Planning systems.
- The research proposes a new approach for feature extraction in ERP systems using machine learning algorithms.
- Future work could involve incorporating additional data sources, such as network traffic logs, to enhance the accuracy of anomaly detection in ERP systems.
- Real-time monitoring systems, integration with existing security systems, and the use of explainable AI techniques are potential directions to further improve the system's effectiveness and usability in detecting and responding to security threats and system errors.
- Overall, the research contributes to the development of multi-layered ERP software architecture with AI decision support, improving fraud detection and enhancing the success of ERP system implementations.

ACKNOWLEDGMENTS

We would like to thank Spor İstanbul for giving us the necessary permission.

AUTHORS' CONTRIBUTIONS

Study concept and design: Ali Gunes. Acquisition of data: Emrah Arslan. Analysis and interpretation of data: Emrah Arslan. Drafting of the manuscript: Emrah Arslan. Critical revision of the manuscript for important intellectual content: Ali Gunes. Statistical analysis: Emrah Arslan. Administrative, technical, and material support: Emrah Arslan, Ali Gunes. Study supervision: Ali Gunes.

CONFLICT OF INTEREST

We would like to inform you that there is no interest conflict among the authors.

REFERENCES

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 Manu-facturing, 2021;32(3), 569-578.
5. Chen, Y., Ren, L., & Zhang, L. Anomaly detection of network traffic based on deep learning in ERP systems. IEEE Access, 2020; 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. Fu-ture 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 Convo-lutional Neural Network. IEEE Access, 2021;9, 55870-55878.
9. Sun J, Zhang Y., Wang Z., Shen L. Fraud Detection in ERP Systems Based on Deep Belief Network. Journal of Computa-tional Science, 2020;41, 101115.
10. Gupta, A. Singh, P. The role of ERP in financial accounting: A systematic review. Journal of Enterprise Information Man-agement,2020; 33(3), 531-554.
11. Krishnan G., Yu, A.S. A comprehensive analysis of ERP adoption drivers, implications, and challenges. Journal of Infor-mation Systems, 2018;32(1), 101-121.
12. Damodaran L. Olafsson, S. Enterprise resource planning systems and financial reporting quality: The role of internal con-trols. 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 Intelligent & 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 detec-tion," 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 planning sys-tems 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. Interna-tional 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 Op-erational 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 Compu-ting, 2015;26, 325-334.
20. Münstermann B., Eckstein J., Kabst R. The use of corporate talent management for succession planning in German medi-um-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 Enter-prise 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:26017442]
23. Sarker, I. H. Machine learning: Algorithms, real-world applications and research directions. SN computer sci-ence, 2021; 2(3), 160. [doi:10.1007/s42979-021-00592-x] [PMid:33778771]
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.