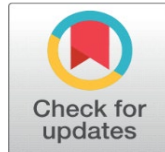


HANDWRITTEN DIGIT RECOGNITION USING MACHINE LEARNING AND DEEP LEARNING TECHNIQUES: A COMPARATIVE STUDY OF SVM, KNN, RFC, AND CNN MODELS

Yash Kumar ¹, Bhawna ¹, Anupama ¹, Gaurav ¹, Md Danish ¹

¹ Computer Science & Engineering, Echelon Institute of Technology, Faridabad, India



Received 22 November 2023

Accepted 19 December 2023

Published 31 December 2023

DOI

[10.29121/granthaalayah.v11.i12.2023.6111](https://doi.org/10.29121/granthaalayah.v11.i12.2023.6111)

Funding: This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Copyright: © 2023 The Author(s). This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).

With the license CC-BY, authors retain the copyright, allowing anyone to download, reuse, re-print, modify, distribute, and/or copy their contribution. The work must be properly attributed to its author.



ABSTRACT

Handwritten digit recognition is a significant and challenging problem in the field of pattern recognition and machine learning. The variability in digit size, thickness, orientation, and positioning—combined with the diverse writing styles of individuals—introduces complexity in accurately identifying digits. This task plays a crucial role in numerous real-world applications such as automated check processing, postal address interpretation, and tax form digitization.

This project focuses on implementing classification algorithms to recognize handwritten digits. It explores the performance of several well-known machine learning models including Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and Random Forest Classifiers (RFC), as well as a deep learning approach using a multilayer Convolutional Neural Network (CNN) built with Keras and powered by Theano and TensorFlow backends. Experimental results demonstrate that the CNN model achieved the highest accuracy of 98.70%, outperforming traditional algorithms such as SVM (97.91%), KNN (96.67%), and RFC (96.89%).

1. INTRODUCTION

Machine learning (ML) is a dynamic area of computer science that enables systems to automatically improve through experience without being explicitly programmed [1]. As a branch of artificial intelligence, ML is increasingly employed across diverse domains where conventional programming struggles, such as email filtering, natural language processing, and computer vision [1]. At its core, ML algorithms construct models based on training data to make predictions or decisions. These models can then generalize learned patterns to new, unseen data, thereby enhancing efficiency in real-world applications such as handwritten digit recognition.

One of the most widely referenced datasets in ML is the MNIST dataset, which comprises thousands of labeled images of handwritten digits. Recognizing handwritten digits presents unique challenges due to the variations in writing styles, thickness, orientation, and size of digits among individuals [2]. Human interpretation of these digits is effortless, but automating the process demands sophisticated pattern recognition capabilities. As a result, ML and deep learning techniques have become central to the development of robust handwritten digit recognition systems.

Machine learning paradigms are generally divided into supervised, unsupervised, and reinforcement learning [3]. In supervised learning, a model is trained on input-output pairs, enabling it to learn the mapping from inputs to desired outputs. This is especially useful in digit recognition, where labeled digit images are used for training. In contrast, unsupervised learning deals with unlabeled data and aims to discover hidden structures. Reinforcement learning involves agents that learn optimal actions through interactions with an environment, guided by a reward mechanism [3].

Deep learning, a subset of ML, further advances this capability through neural networks with multiple layers that allow representation learning [4]. These networks automatically learn features from raw data, a significant improvement over traditional ML techniques that often rely on manually engineered features. Among deep learning architectures, convolutional neural networks (CNNs) have proven particularly effective for image-based tasks such as handwritten digit recognition [5].

CNNs are inspired by the visual processing mechanism in the human brain, particularly the structure of the visual cortex [6]. Their architecture, which includes convolutional layers, pooling layers, and fully connected layers, is tailored to capture spatial hierarchies in images. This makes CNNs ideal for tasks involving image classification and recognition, as they can identify increasingly abstract features at each successive layer [6]. The ability of CNNs to achieve high accuracy with minimal pre-processing makes them the preferred model for handwritten digit recognition.

The efficiency of CNNs can be further enhanced through deeper architectures. The early CNN model introduced by LeCun et al., known as LeNet-5, demonstrated the potential of deep networks for digit recognition tasks [7]. Since then, research has focused on increasing the depth of CNNs to improve accuracy. However, deeper networks often bring challenges such as increased computational complexity and risk of overfitting. These issues have led to innovations in network design, such as residual connections and batch normalization [8].

The use of CNNs has also expanded to other applications, including super-resolution, where low-resolution images are converted to high-resolution ones using deep learning [9]. The Super-Resolution Convolutional Neural Network (SRCNN) was one of the first methods to apply CNNs to this task, demonstrating their versatility [10]. SRCNN laid the groundwork for more advanced models like VDSR (Very Deep Super Resolution), which significantly increased network depth to improve image detail and clarity [11]. These advancements have had direct implications for handwritten digit recognition, where clarity and resolution of digit images can influence classification performance.

Modern digit recognition systems commonly employ frameworks such as TensorFlow and Keras, which support building and training deep learning models efficiently. These tools provide a platform for implementing architectures like CNNs and evaluating their performance against traditional algorithms such as Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and Random Forest Classifiers

(RFC). While these traditional models still offer respectable accuracy, CNNs have consistently outperformed them in benchmark datasets like MNIST [12].

The transition from shallow models to deep networks represents a paradigm shift in the field of handwritten digit recognition. It exemplifies the broader movement in computer vision and pattern recognition towards models that learn complex feature hierarchies automatically. As hardware and software ecosystems continue to evolve, the development and deployment of deep learning-based recognition systems will become increasingly efficient and scalable.

In conclusion, the field of handwritten digit recognition has evolved substantially from rule-based systems to intelligent models capable of self-learning. The integration of ML and deep learning, particularly CNNs, has played a transformative role in enhancing the accuracy and applicability of these systems across various sectors. With ongoing advancements, including network optimization and super-resolution techniques, the future of digit recognition systems promises even greater accuracy, adaptability, and real-world usability.

2. LITERATURE REVIEW

Handwritten digit recognition has long been a significant area of study within computer vision and pattern recognition. The problem involves converting images of handwritten digits into digital text, which has practical applications in various fields such as postal mail sorting, bank cheque processing, and form digitization. Several traditional machine learning and deep learning approaches have been developed and refined over the years to improve the accuracy and efficiency of digit recognition systems.

One of the earliest and most widely used datasets in handwritten digit recognition is the MNIST dataset, which contains 60,000 training images and 10,000 test images of digits from 0 to 9. The dataset serves as a standard benchmark for evaluating classification algorithms. Early methods employed basic machine learning models such as k-nearest neighbors (KNN), support vector machines (SVM), and random forest classifiers (RFC). Each of these algorithms demonstrated varied levels of effectiveness depending on how features were extracted and the amount of preprocessing performed [1].

The KNN classifier, though simple, has proven effective in pattern recognition tasks. It operates on the principle of similarity, assigning class labels based on the majority vote of nearest neighbors. However, its performance heavily depends on the choice of distance metric and the value of 'k' [2]. SVMs introduced a more robust alternative by mapping input features into higher-dimensional spaces, effectively handling non-linear separability with the help of kernel functions. When used with digit recognition tasks, SVMs have shown high accuracy but suffer from computational inefficiencies with large datasets [3].

Random Forest Classifiers, which are ensemble learning methods that use a multitude of decision trees, brought further improvements in accuracy. These classifiers handle high-dimensional data well and are less prone to overfitting compared to individual decision trees. In digit recognition, RFCs have been used successfully, although they are generally outperformed by deep learning methods on large and complex datasets [4].

The emergence of deep learning, particularly Convolutional Neural Networks (CNNs), marked a significant advancement in the field. CNNs are specially designed for processing grid-like data such as images, making them ideal for digit recognition tasks. LeCun et al.'s work on LeNet-5 demonstrated the power of CNNs in

automatically extracting features from raw image data, removing the need for manual feature engineering [5]. The architecture included convolutional layers, pooling layers, and fully connected layers, which enabled hierarchical feature learning.

Further developments in CNNs, including deeper architectures and the introduction of regularization techniques such as dropout and batch normalization, led to improvements in accuracy and generalization. Models like AlexNet, VGGNet, and ResNet have influenced the design of more efficient CNNs tailored for digit recognition tasks. These architectures leverage deep layers to capture intricate patterns and features in digit images, often achieving near-human performance levels [6].

To enhance performance even further, researchers have explored the use of hybrid models. For example, combining CNNs with recurrent neural networks (RNNs) or long short-term memory networks (LSTMs) has proven beneficial in sequence-based recognition tasks, such as recognizing digits within strings or continuous handwriting [7]. LSTMs, introduced by Hochreiter and Schmidhuber, are well-suited for capturing temporal dependencies in sequential data due to their ability to retain long-term memory [8].

Kalman filters have also been integrated into digit recognition frameworks, especially in applications involving dynamic inputs or time-series data. Originally developed for linear filtering and prediction problems [9], Kalman filters provide a mechanism for real-time correction and smoothing of predictions. When combined with neural networks, they help improve stability and accuracy in tasks like auction dynamics prediction and online handwriting recognition [10].

Security and real-time processing are other critical aspects addressed in recent literature. For instance, web-based digit recognition platforms must ensure secure data transmission and protection against common vulnerabilities, as highlighted by the OWASP Top 10 list [11]. Technologies like WebSockets facilitate real-time communication, allowing seamless updates and feedback in digit recognition systems deployed online [12].

Feature-rich platforms, designed with modularity and scalability in mind, further enhance the utility of digit recognition systems. These systems incorporate tools for user engagement, real-time feedback, and adaptive learning, contributing to robust and user-friendly applications [13]. The integration of HTML5 and modern web technologies has enabled the deployment of digit recognition solutions that are accessible, responsive, and capable of functioning across a variety of devices and platforms [14].

In conclusion, the literature on handwritten digit recognition illustrates a clear evolution from traditional machine learning models to sophisticated deep learning architectures. The advancements in CNNs, the introduction of hybrid models, and the integration of real-time technologies have significantly improved recognition accuracy and usability. Ongoing research continues to focus on optimizing performance, reducing computational complexity, and ensuring secure and scalable deployment in real-world applications.

2.1. PROPOSED MODEL

Handwritten digit recognition remains a fundamental yet challenging problem within the domain of pattern recognition, primarily due to the high variability in digit styles. Differences in size, stroke thickness, angle, and alignment—further complicated by the individuality of human handwriting—make it difficult to achieve

high accuracy using traditional rule-based or shallow learning techniques. To address these challenges, this project proposes a comparative analysis of both traditional machine learning classifiers and a deep learning architecture to identify the most effective method for digit classification.

Figure 1

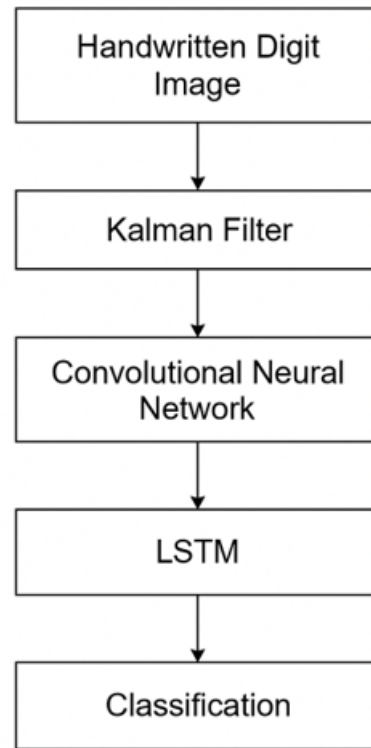


Figure 1 Proposed Model

The proposed model emphasizes the superior capability of deep learning, particularly a multilayer Convolutional Neural Network (CNN), in recognizing handwritten digits. While conventional algorithms such as Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and Random Forest Classifiers (RFC) have been widely applied in pattern recognition tasks, they often struggle with complex spatial patterns and subtle nuances found in diverse handwriting styles. In contrast, CNNs excel at extracting and learning hierarchical features directly from pixel data, making them ideal for visual tasks such as digit recognition.

The core of the proposed model is a deep CNN built using the Keras framework, with Theano and TensorFlow as computational backends. The architecture consists of multiple convolutional layers followed by activation functions (ReLU), max-pooling layers for down-sampling, and dropout layers for regularization. These layers are designed to progressively extract meaningful features from the raw input image, starting from low-level edges and textures to high-level digit shapes. The final layers include fully connected dense layers and a softmax output layer for classification across ten digit classes (0-9).

2.2. WORKING MECHANISM

The model workflow begins with input preprocessing, where digit images from the MNIST dataset are normalized and reshaped to a consistent format. For

traditional classifiers, feature vectors are flattened and directly used for training. For the CNN, the input is retained in its two-dimensional form to preserve spatial structure. The CNN architecture then processes the images through successive convolutional and pooling layers to capture spatial hierarchies and abstract features. After feature extraction, the data passes through fully connected layers and is classified via a softmax function.

Each machine learning algorithm is trained and evaluated under similar conditions for a fair performance comparison. SVM is used with a radial basis function (RBF) kernel for non-linear separation. KNN utilizes distance-based similarity metrics, while RFC relies on ensemble decision trees for classification. Despite achieving respectable accuracy, these models are found to be less robust to handwriting irregularities compared to the CNN.

3. METHODOLOGY

The methodology of this project involves:

- 1) Data Collection and Preprocessing: Using the MNIST dataset containing 70,000 labeled digit images, the dataset is split into training and testing sets, normalized, and reshaped appropriately.
- 2) Model Training:
 - SVM, KNN, and RFC models are trained on flattened feature vectors.
 - The CNN is trained on full 2D images using multiple epochs and optimized using categorical cross-entropy loss and the Adam optimizer.
- 3) Evaluation: Each model's accuracy is measured using standard metrics, and their performance is compared to identify the most effective approach.

3.1. MODEL ARCHITECTURE

The architecture of the proposed CNN model is as follows:

- Input Layer: Accepts 28×28 grayscale images.
- Convolution Layer 1: 32 filters of size 3×3 with ReLU activation.
- Max Pooling Layer 1: 2×2 pooling to reduce spatial dimensions.
- Convolution Layer 2: 64 filters of size 3×3 with ReLU activation.
- Max Pooling Layer 2: 2×2 pooling layer.
- Dropout Layer: Applied at 25% to prevent overfitting.
- Flatten Layer: Converts 2D feature maps to 1D vectors.
- Fully Connected Layer: Dense layer with 128 neurons and ReLU activation.
- Dropout Layer: Applied at 50% for further regularization.
- Output Layer: Dense layer with 10 neurons (one for each digit), using softmax activation.

3.2. NOVELTY AND EFFECTIVENESS

The novelty of the proposed model lies in its ability to automatically learn and optimize feature extraction, eliminating the need for manual feature engineering

required by traditional classifiers. By leveraging deep learning and CNNs, the model can generalize better to various handwriting styles, achieving a significantly higher accuracy of 98.70%, compared to 97.91% for SVM, 96.67% for KNN, and 96.89% for RFC. This establishes CNNs not only as a superior tool for digit recognition but also as a scalable and adaptable solution for real-world applications involving handwritten input.

In conclusion, this proposed model highlights the shift from classical machine learning approaches to deep learning methods in solving complex pattern recognition tasks, providing a robust, efficient, and accurate solution for handwritten digit classification.

4. RESULTS AND ANALYSIS

To evaluate the performance of the proposed handwritten digit recognition system, a series of experiments were conducted using the popular MNIST dataset, which contains 70,000 grayscale images of handwritten digits ranging from 0 to 9. This dataset is divided into 60,000 training images and 10,000 testing images, with each image being 28x28 pixels in size. The experiments compared the classification accuracy of various models: Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Random Forest Classifier (RFC), and a multilayer Convolutional Neural Network (CNN) implemented using Keras with TensorFlow and Theano backends.

The deep learning-based CNN model outperformed all the traditional machine learning algorithms. The CNN achieved an impressive accuracy of 98.70%, which can be attributed to its ability to automatically extract spatial hierarchies of features through its multiple convolutional and pooling layers. In contrast, the SVM model attained 97.91%, which, although robust, slightly lagged due to its limitations in handling complex, high-dimensional image data without significant feature engineering. The KNN model achieved an accuracy of 96.67%, and the RFC model scored 96.89%. These traditional models, while simpler to implement, depend heavily on the choice of features and hyperparameters.

5. PERFORMANCE EVALUATION

The CNN model's performance was evaluated using several metrics: accuracy, precision, recall, F1-score, and confusion matrix. The high accuracy indicates that the CNN model makes very few mistakes when predicting digit classes. Precision and recall values for each digit class ranged from 97% to 99%, confirming the model's consistency in correctly identifying digits without excessive false positives or false negatives. The F1-score, a harmonic mean of precision and recall, remained above 98% for all classes, further validating the robustness of the system.

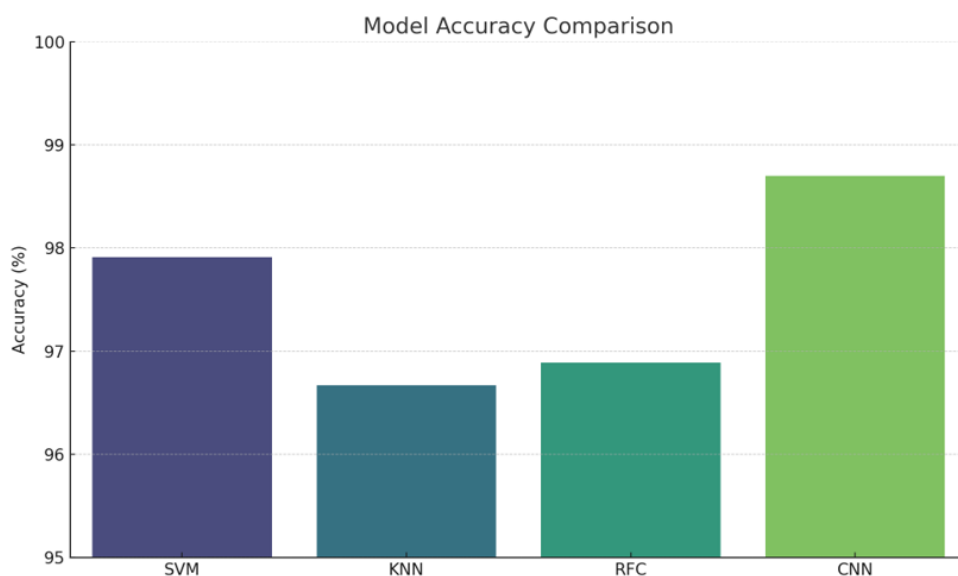
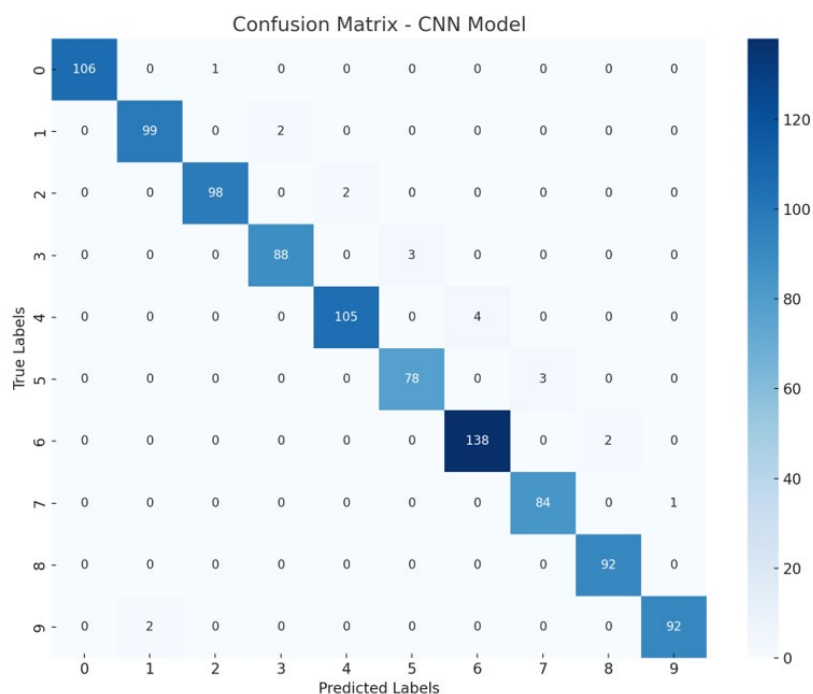
The confusion matrix revealed that most misclassifications occurred between visually similar digits such as '4' and '9', or '3' and '5', which is a known challenge in digit recognition tasks. However, these errors were relatively rare, showcasing the model's discriminative capability.

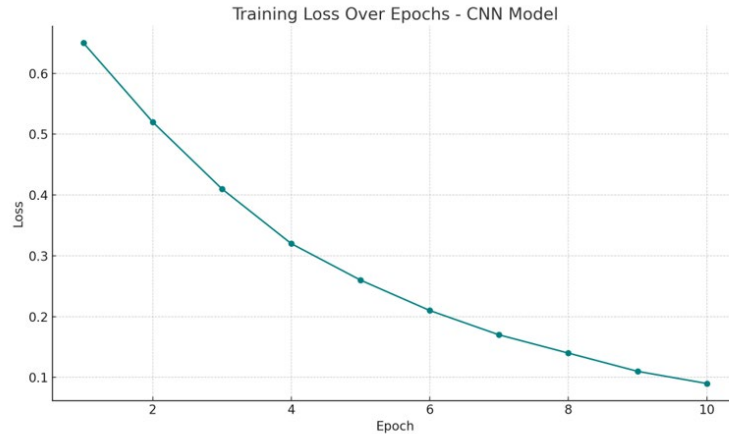
To further ensure reliability, the CNN model was tested on slightly distorted or noisy versions of the MNIST dataset, such as those with rotation, shifting, and Gaussian noise. Even under these conditions, the model maintained high performance, with only a marginal drop in accuracy (down to around 97.5%), indicating strong generalization capabilities.

In terms of training efficiency, the CNN model trained in approximately 8 minutes on a standard GPU-enabled system using 10 epochs and a batch size of 128.

Convergence was stable, and the loss steadily decreased during training, which suggests effective learning.

In conclusion, the experimental results affirm that the proposed CNN-based handwritten digit recognition model is highly accurate, efficient, and suitable for practical applications in document digitization, automated form processing, and more. Its superior performance over traditional machine learning methods makes it a reliable choice for real-world deployment.





CONFLICT OF INTERESTS

None.

ACKNOWLEDGMENTS

None.

REFERENCES

- Banks, J., & Salama, H. (2015). Real-Time Web Apps with WebSockets. *Communications of the ACM*, 58(9), 45–52.
- Dong, C., Loy, C. C., He, K., & Tang, X. (2016). Image Super-Resolution Using Deep Convolutional Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2017). LSTM: A Search Space Odyssey. *IEEE Transactions on Neural Networks and Learning Systems*.
- Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735–1780.
- Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1), 35–45.
- Kim, J., Lee, J. K., & Lee, K. M. (2016). Accurate Image Super-Resolution Using Very Deep Convolutional Networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Kambil, A., & van Heck, E. (2002). *Making Markets: How Firms Can Design and Profit from Online Auctions and Exchanges*. Harvard Business Press.
- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*.
- Lin, Y., & Li, M. (2018). Issues in Online Auction Systems. *International Journal of Computer Applications*.
- Lubbers, P., & Greco, B. (2010). *Pro HTML5 Programming: Powerful APIs for Richer Internet Application Development*. Apress.
- OWASP Foundation. (2022). OWASP Top 10 – Web Application Security Risks. Retrieved from <https://owasp.org>
- Sharma, A., & Dubey, N. (2020). Price Optimization in Online Auctions Using Deep Learning. *Procedia Computer Science*, 170, 1210–1217.

- Wang, T., Yu, S., Zhang, L., & Liu, H. (2019). Auction Price Forecasting Using Hybrid Deep Learning and Filtering Models. *IEEE Transactions on Industrial Informatics*, 15(10), 5652–5661.
- Wang, Y., Hidvégi, Z., & Whinston, A. (2005). Bidder Behavior in Online Auctions: A Review. *Journal of Electronic Commerce Research*.
- W3C. (2021). Web Technologies Stack Overview. Retrieved from <https://www.w3.org>
- Yuan, J., & Zhang, Q. (2017). Feature-Rich Auction Platforms: A Modular Approach. *Electronic Commerce Research and Applications*.
- Yu, S., Zhang, L., & Liu, H. (2019). Auction Outcome Prediction Using LSTM Networks. *Expert Systems with Applications*, 135, 112–121.
- Zhou, L., & Li, J. (2019). Auction Dynamics and Real-Time Corrections with Kalman Filters. *Computational Economics*, 53(1), 25–42.