






Article

Fine-Tuning Artificial Neural Networks to Predict Pest Numbers in Grain Crops: A Case Study in Kazakhstan

Galiya Anarbekova¹, Luis Gonzaga Baca Ruiz^{2,*} , Akerke Akanova^{1,*}, Saltanat Sharipova³ 
and Nazira Ospanova⁴ 

¹ Department of Computer Science, S. Seifullin Kazakh Agrotechnical Research University, Astana 010011, Kazakhstan; galiya.anarbekova@kazatu.edu.kz

² Department of Software Engineering, University of Granada, 18014 Granada, Spain

³ Department of Computer Science, Astana Information Technology University, Astana 010000, Kazakhstan; saltanat.sharipova@astanait.edu.kz

⁴ Department of Computer Science, S. Toraihyrov Pavlodar State University, Pavlodar 140008, Kazakhstan; ospanova.n@teachers.tou.edu.kz

* Correspondence: bacaruiz@ugr.es (L.G.B.R.); a.akanova@kazatu.edu.kz (A.A.)

Abstract: This study investigates the application of different ML methods for predicting pest outbreaks in Kazakhstan for grain crops. Comprehensive data spanning from 2005 to 2022, including pest population metrics, meteorological data, and geographical parameters, were employed to train the neural network for forecasting the population dynamics of *Phyllotreta vittula* pests in Kazakhstan. By evaluating various network configurations and hyperparameters, this research considers the application of MLP, MT-ANN, LSTM, transformer, and SVR. The transformer consistently demonstrates superior predictive accuracy in terms of MSE. Additionally, this work highlights the impact of several training hyperparameters such as epochs and batch size on predictive accuracy. Interestingly, the second season exhibits unique responses, stressing the effect of some features on model performance. By advancing our understanding of fine-tuning ANNs for accurate pest prediction in grain crops, this research contributes to the development of more precise and efficient pest control strategies. In addition, the consistent dominance of the transformer model makes it suitable for its implementation in practical applications. Finally, this work contributes to sustainable agricultural practices by promoting targeted interventions and potentially reducing reliance on chemical pesticides.

Keywords: artificial neural networks; bread striped flea; agriculture; forecasting



Citation: Anarbekova, G.; Ruiz, L.G.B.; Akanova, A.; Sharipova, S.; Ospanova, N. Fine-Tuning Artificial Neural Networks to Predict Pest Numbers in Grain Crops: A Case Study in Kazakhstan. *Mach. Learn. Knowl. Extr.* **2024**, *6*, 1154–1169. <https://doi.org/10.3390/make6020054>

Received: 28 March 2024

Revised: 21 May 2024

Accepted: 23 May 2024

Published: 26 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Accurate pest prediction is a crucial issue in agriculture, empowering farmers to proactively manage and mitigate pest infestations, minimize crop damage, and enhance yields. The origin of precision agriculture, which incorporates advanced technologies such as remote sensing, data analytics, and integrated pest management systems, is instrumental in facilitating early pest detection. This equips farmers with the capability to make informed, data-driven decisions. By integrating pest data with other relevant sources such as weather conditions and soil moisture, farmers can develop predictive models to anticipate pest outbreaks and implement timely interventions, thereby reducing the reliance on chemical pesticides and embracing more sustainable solutions [1,2]. Additionally, precision farming technologies, including decision support systems, are instrumental in predicting potential pest outbreaks and optimizing the use of equipment, further emphasizing the significance of effective pest prediction in agriculture [3,4]. The substantial benefits of precision agriculture extend beyond mere pest mitigation as it promotes data-driven decision making and seamlessly integrates pest management into broader farm management strategies.

The importance of effective pest prediction in agriculture is strongly related to the pivotal role that grain crops play in global food security. Grain crops, including cereals,

legumes, and oilseeds, serve as essential food sources for a substantial portion of the world's population [5]. However, the threat posed by pest infestations to these crops is significant, resulting in considerable economic losses and jeopardizing global food security. Estimates suggest that between 20% and 40% of global crop production is annually lost to pests, imposing staggering economic costs [6]. Plus, pests such as aphids and the viruses they transmit can inflict severe damage to wheat, causing substantial economic losses for growers. Moreover, beyond direct crop damage, pest infestations in stored grains lead to significant losses, impacting the nutritional value and quality of the food supply.

Among the challenges faced by pests in farming, there is a need to explore new technological and intelligent solutions. Thus, technologies such as artificial neural networks (ANNs), remote sensing, and data analytics provide a fresh way to deal with pest issues in grain crops [7–9]. These cutting-edge technologies not only enhance early detection capabilities but also enable farmers to make informed decisions based on comprehensive data analyses. As we step into it, using these tools becomes a strong approach to changing how we predict and manage pests, making farming more sustainable and effective.

In this context, ANNs have emerged as powerful tools for addressing complex problems, including pest prediction in agriculture [7,10]. ANNs are widely adopted for predicting crop pest risks because of their ability to effectively model and analyze intricate relationships within large datasets. Research has demonstrated the high prediction accuracy of ANNs in forecasting pest populations and quantifying their risks in various crops, such as rice [11]. In addition, ANNs have been applied in intelligent agent-based prediction systems for pest detection and alert mechanisms, leveraging technologies such as acoustic approaches and video processing techniques to enable early pest discovery and classification [12].

In Kazakhstan, where grain crops hold immense economic importance as the largest grain producer in Central Asia [13,14], a connection between the vital role of agriculture and the imperative for effective pest management becomes evident. Since farming, especially growing grains, is a big part of the country's economy, society, and environment, more than a third of the people depend on agriculture for their jobs and way of life. While wheat dominates as the primary crop, constituting 80% of Kazakhstan's grain production and serving as a key contributor to economic growth and trade, the sector faces challenges due to limited modernization and vulnerability to weather conditions [15]. The significance of grain crops for food security and the economy in Kazakhstan underscores the need for effective pest management strategies to safeguard production and mitigate the potential economic impact of pest infestations.

As a consequence, the motivation behind focusing on fine-tuning ANNs for pest prediction in grain crops lies in the pursuit of optimizing predictive accuracy and enhancing the applicability of these advanced models [11,16]. ANNs, known for their ability to capture complex patterns within data, offer a promising tool to address the difficulty of pest dynamics. Fine-tuning, in this context, becomes essential to systematically adjust the model's hyperparameters, such as the number of layers and neurons, to achieve an optimal configuration. By adapting the architecture of ANNs, we aim to uncover the most effective configuration that maximizes predictive performance [17], ensuring that the models are finely tuned to the unique characteristics of pest data in grain crops. This research has a two-fold objective: to refine pest prediction methods and to promote sustainable agriculture by enabling more precise interventions, thereby reducing dependence on chemical pesticides.

Literature Review

In the realm of agriculture, the use of ANN significantly enhances the efficiency of plant cultivation. The application of these technologies enables the monitoring of plants, forecasting crop yields, predicting weather and climate changes, optimizing resource utilization, and combating diseases and pests. For instance, researchers from the University of Barishal (Bangladesh) employed deep convolutional neural networks (CNN) to detect tomato leaf diseases [18]. Gaussian and median filters were applied, two color models (HSI

and CMYK) were utilized, and images were processed in four different ways. CNNs have also been integrated into the work of other researchers from the University of Ségou in Mali [19]. They proposed a method for detecting and identifying insect pests in crops using a CNN. The network parameters were reduced by 58.90%, making it effective in agricultural crop diagnostics. Similar promising results with CNN have been achieved by researchers from China in their study for recognizing diseases and pests in strawberries [20]. They employed pre-trained parameters from the original AlexNet model and adopted the Adam optimization algorithm for parameter updates to expedite the training process, achieving an accuracy of 97.35%.

To swiftly identify prevalent pests in agricultural and forestry settings, a technique leveraging a deep CNN was introduced in [21]. The authors used image processing methods, including flipping, rotation, and scaling, as well as the incorporation of Gaussian noise and unconventional principal component analysis (PCA) to enhance the effectiveness of the approach.

ANNs have proven to be versatile tools not only in image classification but also useful for tackling other different kinds of problems. For example, scientists have used a combination of CNN and recurrent neural networks (RNN) to predict crop yield [22]. In this study, a CNN-RNN framework was employed to forecast corn and soybean yields across the entire Corn Belt in the United States. Another popular predictive model, long short-term memory (LSTM), was used in [23] to predict future occurrences of pests and diseases in cotton based on weather conditions. In this study, the Apriori algorithm was used to reveal relationships between weather factors and the emergence of cotton pests. The authors developed an LSTM-based method to tackle challenges in time series forecasting. LSTM exhibited proficiency in predicting the occurrence of pests and diseases in cotton fields. Scientists from Guangxi University in China have developed a model for predicting the pressing process index of sugarcane based on an ANN [19]. This model integrates deep learning with the pressing mechanism and is constructed using a physics-based neural network to forecast the sugarcane pressing process index.

Another lesser-known model is the Multi-Task Neural Network (MTNN). It is a deep learning technique that can perform multiple tasks simultaneously, such as semantic segmentation and edge detection. MTNN has been applied to various agricultural problems, such as fruit freshness classification, field boundary extraction, and crop disease detection. Luo et al. [24] proposed a multi-task learning network with an attention-guided mechanism for segmenting agricultural fields from high-resolution remote sensing images. Their method achieved high accuracy and improved the completeness of field boundaries. Kang and Gwak [25] developed an ensemble model that combines the bottleneck features of two MTNNs with different architectures for fruit freshness classification. Their model outperformed other transfer learning-based models and showed effectiveness in distinguishing fresh and rotten fruits. Another example can be found in [26], where the authors introduced a semantic edge-aware MTNN to obtain closed boundaries when delineating agricultural parcels from remote sensing images. Their method improved conventional semantic segmentation methods and handled small and irregular parcels better.

Literature analysis reveals a notable gap in the exploration of neural networks for pest detection in the agricultural sector. This area calls for more comprehensive research, particularly in addressing pests such as the wheat stripe flea beetle. The wheat stripe flea beetle stands out as the primary pest affecting crops in the northern and central regions of Kazakhstan (*Phyllotreta vittula*), causing substantial damage to all grain crops, and impacting overall yield. In agriculture, the timely identification of pests is a significant challenge exacerbated by seasonal variations, extreme weather conditions, and global climate change, all of which contribute to the risk of pest outbreaks.

This research aims to fill this gap by developing an ANN-based model dedicated to predicting the dynamics of grain crop pests, ultimately enhancing crop yield. To accomplish this objective, two primary tasks have been outlined: (a) selecting the optimal ANN

architecture for predicting pest dynamics, and (b) determining the optimal hyperparameters for the chosen neural network architectures.

2. Methodology

This section introduces the proposed methodology to predict the dynamics of the number of pests on grain crops in Kazakhstan. We describe here the dataset with some statistics and the preprocessing methods applied, including normalization and missing value imputation. Finally, the three models implemented are briefly introduced: MLP, MTNN, and LSTM.

2.1. Dataset

To train a neural network for forecasting the population dynamics of pests, specifically *Phyllotreta vittula* in Kazakhstan, comprehensive data spanning from 2005 to 2022 were employed. The dataset incorporates crucial information, encompassing both pest population metrics and meteorological data. The chosen input features encompass diverse parameters: The year (year), geographical location (location), and soil surface temperature from April to September (t_{apr} , t_{may} , t_{jun} , t_{jul} , t_{aug} , t_{sep}). Additionally, precipitation data from April to September (p_{apr} , p_{may} , p_{jun} , p_{jul} , p_{aug} , p_{sep}) and surveyed area during spring, summer, and autumn (a_{sp} , a_{su} , a_{au}) are integrated as input features. The neural network is trained to predict infestation levels by cereal leaf beetles during spring, summer, and autumn (i_{sp} , i_{su} , i_{au}) as the target variables. Some statistical metrics can be found in Table 1.

Table 1. Descriptive statistics of the dataset, encompassing pest population dynamics (*Phyllotreta vittula*) and meteorological factors from 2005 to 2022 in Kazakhstan.

Variable	Mean	Std	Min	25%	50%	75%	Max
t_{apr}	7.2	3.9	−0.6	4.5	6.7	9.3	19.4
t_{may}	16.8	5.7	1.2	15.6	17.5	19.8	28.3
t_{jun}	23	6.1	2.2	22	24.2	26.4	32.9
t_{jul}	25	4.8	2.8	23.8	25.5	27.2	34.8
t_{aug}	21.2	7.1	2.1	21.2	22.9	25.4	31.9
t_{sep}	14.3	3.4	1.3	12.8	14.4	16	23.8
p_{apr}	0.4	0.3	0.1	0.1	0.3	0.6	1
p_{may}	0.4	0.3	0.1	0.2	0.3	0.6	1
p_{jun}	0.4	0.2	0.1	0.2	0.3	0.5	1
p_{jul}	0.4	0.2	0.1	0.2	0.3	0.5	1.3
p_{aug}	0.4	0.2	0.1	0.2	0.3	0.5	1
p_{sep}	0.4	0.3	0.1	0.2	0.3	0.6	1
a_{sp}	1.5	3.8	0.1	0.2	0.5	52.6	52.6
a_{su}	9.2	13.8	0.1	0.2	14.3	14.3	113.5
a_{au}	7.1	11.6	0.1	0.2	1.1	11.2	131.5
i_{sp}	0.8	1.2	0.1	0.2	0.5	1	9.3
i_{su}	6.5	10.9	0.1	0.2	1	56.8	56.8
i_{au}	4.6	7.3	0.1	0.2	0.7	6.4	32.4

The information presented in Table 1 is fundamental to gaining insights into the dataset used to train the ANN models. The table shows that the variables have diverse ranges, for instance, with areas in autumn and summer, a_{au} and a_{su} respectively, having higher maximum values, while the temperature in April, t_{apr} has a lower minimum value. The distribution of each variable is presented in the table, providing a comprehensive overview of the dataset's characteristics. These trends and variations are further visualized in Figure 1, where the statistical ranges, represented by the maximum and minimum values, provide a clear visual representation of the dataset's distribution. This combination of tabular and visual representations enhances our understanding of the dataset's dynamics.

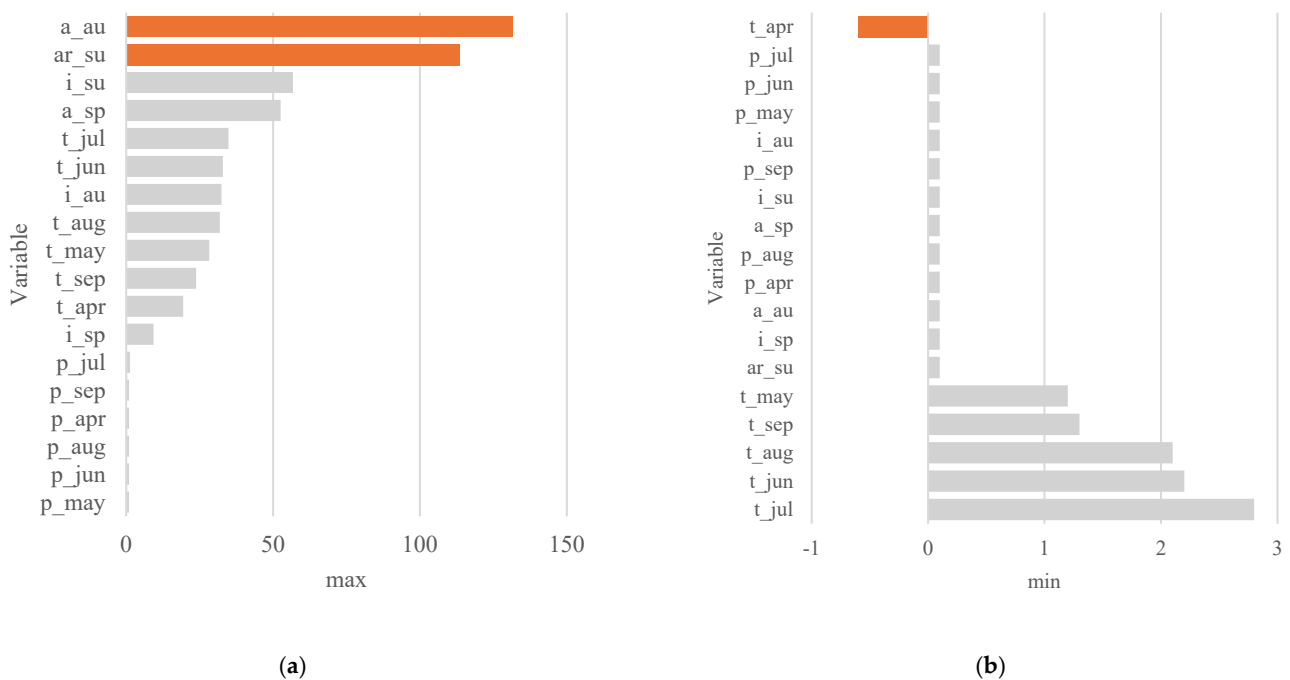


Figure 1. Distribution of minimum (a) and maximum (b) values for each variable in the dataset.

The intricate and multifaceted nature of the pest population data in Kazakhstan requires the use of robust models. To address this challenge, we have chosen three well-established ANN architectures: Multi-layer perceptron (MLP), multi-task neural network (MTNN), long short-term memory (LSTM), and transformer. These models are renowned for their adaptability and proficiency in capturing complex data patterns, making them ideal for forecasting pest population dynamics in this region.

2.2. Preprocessing

Before applying any machine learning (ML) techniques, we performed two preprocessing methods to guarantee the quality and validity of the data. The first method was to apply the Standard Scaler transformation to normalize the numerical features and reduce the effect of outliers. The second method was to handle the missing values in the dataset by imputing them with appropriate values based on the feature type and distribution.

First, the data were standardized using the Standard-Scaler method before performing any statistical analysis. This preprocessing step ensured that the different features in the dataset had a consistent and comparable scale. The StandardScaler method adjusted the data distribution to have a zero mean and a unit standard deviation, thus minimizing the influence of outliers and scale differences among different features. By standardizing the data, the model training process became more stable and efficient, as the optimization algorithm could converge faster and better. Moreover, standardization often helps to improve the performance of certain machine learning algorithms, leading to more accurate parameter estimation and better prediction results.

The missing data were replaced with the median value due to its robustness to outliers and its ability to preserve the central tendency of the data distribution. Using the median helps mitigate the impact of extreme values or outliers present in the dataset, which could disproportionately influence the mean value. By replacing missing values with the median, the integrity of the dataset is maintained, and the risk of introducing bias into the analysis is reduced.

2.3. Predictive Models

Our primary focus is on developing and analyzing various ANN models for pest detection and forecasting. However, we also incorporated support vector regression (SVR)

into our analysis. We consider this model because SVR has distinct characteristics and strengths that make it useful in handling different types of data and patterns. Specifically, SVR performs well in situations where the relationship between inputs and outputs is nonlinear or when working with smaller datasets. By including SVR in our comparison, we provide a wider glimpse of different machine learning techniques and assess their suitability for the given task.

A. Support Vector Regression

SVR, a machine learning technique based on support vector machines, is adapted for predicting continuous values rather than class labels. SVR works by finding the hyperplane that best fits the data, while maximizing the margin, to minimize the error between the predicted and actual values. This approach makes SVR particularly effective for tasks where the number of features is much greater than the number of samples [27].

B. Multi-Layer Perceptron

MLP is a type of ANN that has at least three layers of nodes that are connected to each other: An input layer, an output layer, and one or more hidden layers in between [28]. MLP can learn complex patterns in data by using a process of forward and backward propagation, where it passes the data through the layers and adjusts the weights of the connections based on the errors. MLP is useful for solving problems that require nonlinear transformations of the input data.

C. Multi-Task Neural Network

Multi-task learning is a subfield of ML that aims to improve the performance of multiple learning tasks by exploiting their commonalities and differences [25]. It can be seen as a form of inductive transfer, where the knowledge learned from one or more related tasks can help generalize better on a new task. It can also provide regularization and data augmentation benefits, as well as enable fast adaptation to new domains and tasks. Unlike traditional neural networks with a single output layer, MTNN incorporates multiple output layers, each dedicated to a distinct task. This architecture facilitates the simultaneous learning of multiple tasks, enhancing efficiency and performance in scenarios where tasks are interrelated or complementary.

D. Long Short-Term Memory Neural Network

LSTM [29,30] is a type of RNN that can learn long-term dependencies in sequential data. Unlike standard RNNs, LSTM has a memory cell that can store and update information over time, avoiding the problem of vanishing or exploding gradients. LSTM can handle complex tasks such as machine translation, speech recognition, and text generation, among others. LSTM consists of three gates: input, output, and forget, that control how information flows in and out of the memory cell.

As an example, the architecture of the LSTM model is depicted in Figure 2. Within an input layer, neurons encode various features such as year, location, soil surface temperature, and precipitation, along with the surveyed area in spring, summer, and autumn, respectively. The rectified linear unit (ReLU) activation function, f , is utilized in both the LSTM and dense layers. The outputs of these layers are subsequently channeled into specific layers designated for each season (spring, summer, and autumn). Following transformation, the outputs are forwarded to the corresponding output layers.

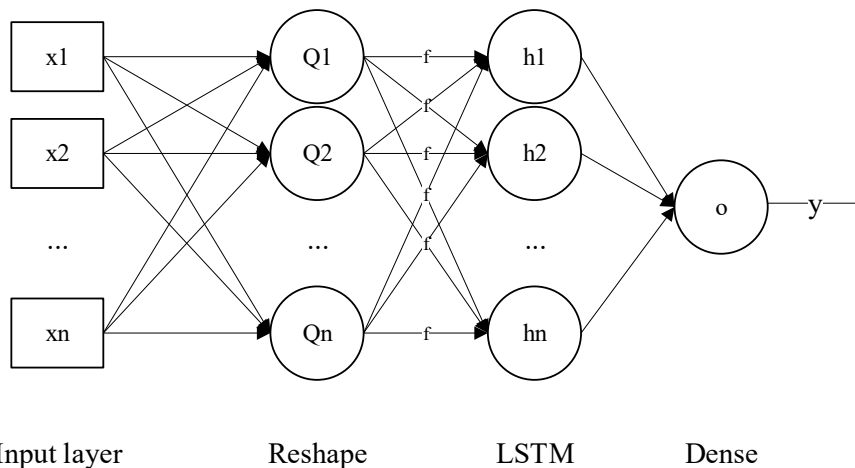


Figure 2. Representation of the LSTM architecture implemented.

E. Transformer

The transformer is a type of neural network architecture (see Figure 3) widely used in natural language processing and sequential data tasks. Unlike traditional neural networks, Transformers rely on self-attention mechanisms to weigh the importance of different parts of the input data, enabling them to capture long-range dependencies more effectively. This mechanism allows transformers to process sequences of varying lengths in parallel, making them computationally efficient. Transformers have demonstrated superior performance in tasks such as language translation, text generation, and sequential data analysis, thanks to their ability to model complex relationships and dependencies within sequences [31].

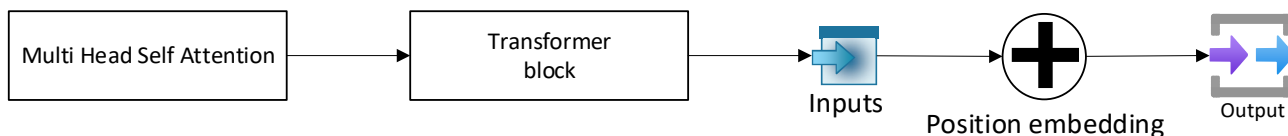


Figure 3. Representation of the transformer architecture implemented.

The multi-head self-attention layer implements the attention mechanism used in transformers. It computes attention weights between each element in the sequence, incorporating sub-layers for query, key, and value projections, and consolidates the results into a single output. Following this, the transformer block layer performs various operations, including feature transformation, dropout to prevent overfitting, and normalization to stabilize training. The input layer is where the model receives the time series data, while the position embedding layer adds crucial information about the sequence’s order. Finally, the output layer represents the model’s prediction for the input sequence.

ReLU offers several advantages, such as computational simplicity and mitigation of the vanishing gradient problem. It contributes to the effective training of deep neural networks. Graphically, the ReLU function can be represented as a line passing through the origin and then rising to the right. It stands out as one of the most popular activation functions in deep learning.

2.4. Hyperparameter Tuning

Grid search cross-validation [32] is a widely utilized tool in the field of ML, specifically designed for the crucial task of hyperparameter tuning. Hyperparameters, which exert control over a model’s learning process, encompass variables such as the number of layers in a neural network, learning rate, or regularization parameter. Selecting optimal values for these parameters holds paramount importance in maximizing a model’s overall performance.

The operational framework of grid search cross-validation follows a systematic approach. Initially, a grid of hyperparameter values is defined, specifying potential ranges for each parameter to be tuned. Subsequently, the method iterates through this grid, systematically evaluating various combinations of hyperparameter values. For each combination, it generates a new instance of the model, conducts training on data subsets using cross-validation, and assesses performance metrics based on a chosen evaluation criterion. After assessing all combinations, the method identifies the set of hyperparameters that result in the most optimal performance.

For our specific dataset, GridSearchCV is a more suitable method for hyperparameter tuning compared to evolutionary computation [33], swarm algorithms [34], and the combined approach using artificial bee colony and differential evolution algorithms [35].

The motivation lies in the relatively small size of the dataset and the limited number of hyperparameters, enabling us to conduct an exhaustive search of hyperparameter combinations using GridSearchCV without significant computational costs. This approach ensures coverage of the entire search space and can provide precise results for optimal model tuning on this dataset. While evolutionary and swarm algorithms may be more efficient for larger or more complex datasets with larger hyperparameter spaces, GridSearchCV remains the preferred choice in this context.

3. Experiments

The selection of hyperparameters is crucial for the success of training ANN. Random search for hyperparameters in neural networks can take a considerable amount of time to converge and yield good results, given the substantial time required for training networks with many parameters.

Determining optimal hyperparameters for the final architecture involved careful experimentation with different configurations, covering various layer numbers and neuron counts. Parameters such as the number of layers and neurons, the number of training epochs, batch size, and the choice of optimizer are crucial when tuning neural network hyperparameters. These parameters significantly influence the model's ability to learn and generalize to new data.

To formulate network architectures, it is necessary to determine the optimal number of layers and neurons in each layer. Insufficient or excessive layer count has corresponding consequences: It may fail to capture important patterns or lead to overfitting. Thus, a balance between learning and preventing overfitting needs to be achieved. Insufficient training epochs may result in inadequate training, while an excessive number of epochs can increase training time without substantial performance improvement.

The general process followed to train our models can be seen in Figure 4. The input data—temperature, precipitation, location, year, and surveyed area—are processed and properly prepared. Then a set of hyperparameters is selected to be optimized: number of layers, neurons, epochs, and batch size. There are many ways to automatically select hyperparameters. The most natural way to select hyperparameters is to iterate over a grid with sliding control. During the experimental stage, hyperparameter tuning was conducted by integrating grid search cross-validation. The input data were split into training and testing sets, comprising 80% and 20% of the data, respectively. Thus, the training set is used as input for training the ANN model, while the test set is utilized to evaluate the model's performance on unseen data.

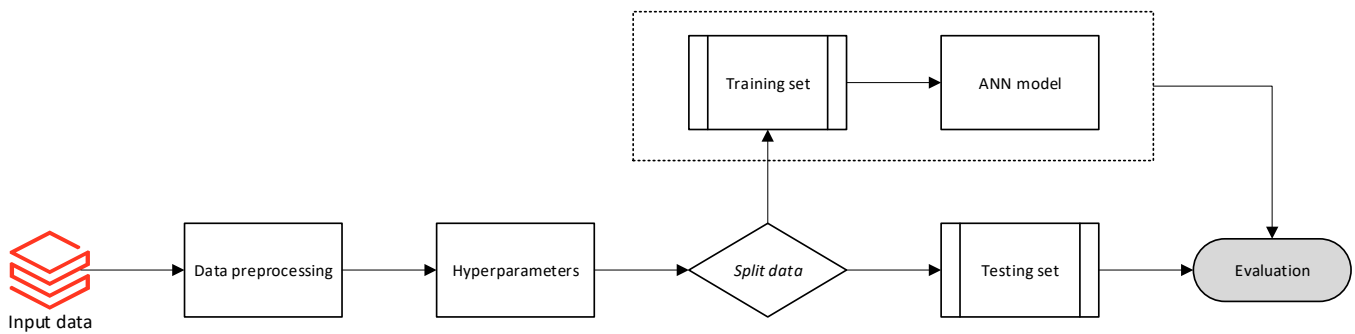


Figure 4. General scheme of the proposed learning process.

All the models implemented were based on the high-level API for building and training deep learning models in the Tensorflow Keras library.

In this study, the model's performance on the training and testing datasets was assessed using statistical parameters, including mean squared error. Compared with other error measures, MSE is differentiable and effective using mathematical methods [36].

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2, \quad (1)$$

where n —the number of observations, y_i —the actual value of the dependent variable for the i th observation, \hat{y}_i —the predicted value of the dependent variable for the i th observation.

In our context, MSE holds significance in evaluating model performance, particularly in regression problems. MSE prioritizes accurate forecasts by penalizing larger errors more heavily. This results in a single, easy-to-understand metric for the models' comparison. Moreover, its mathematical form allows for efficient optimization algorithms to fine-tune models and achieve the best possible performance.

It is important to note that our experiments are not carried out in isolation but rather are interrelated. Each experiment relies on the best outcomes of the previous one. Our methodology results in choosing the most effective parameters in one experiment and subsequently applying them in successive iterations. This iterative cycle enables us to enhance our outcomes incrementally.

4. Results

This section includes the MSE analysis of diverse predictive models under several experiments. The adjusted hyperparameters are the number of neurons, layers, epochs, and batch size. The MSE analysis is displayed in tables organized by season, with each season featuring three columns dedicated to each predictive model for easy performance comparison. We would like to mention that while statistical tests are important to evaluate the significance of differences between results, in our study, we did not include any statistical tests. This decision was based on the observation that the results obtained from the different models exhibited significant variations, indicating obvious differences in performance among them. As such, conducting statistical tests to compare these results would not have yielded meaningful insights, as the differences were already apparent.

Table 2 presents the results of the experiment with different neurons. Importantly, season 3 yielded the most favorable outcomes, especially with the transformer model, which achieved the lowest error rate across all seasons. Season 2, however, proved to be the most challenging for modeling, as evidenced by its highest error rate. The transformer model consistently performed well, achieving the lowest error rates in all seasons. In addition, the table did not reveal a clear trend of improvement in model performance with increasing the number of neurons across all seasons. Notably, only the LSTM model in season 2 and transformer in season 3 showed some improvement with more neurons. Interestingly, the MTNN model achieved its best performance with the lowest number of

neurons (2) in both season 2 and season 3. While MTNN consistently produced reliable results with low variability within each experiment, its overall MSE performance did not surpass that of other models. Finally, comparing MLP and LSTM models, their results were very similar, with LSTM generally achieving a slightly lower MSE. This proximity in performance suggests their competitiveness, but the transformer emerged as the slightly more favorable choice based on predictive accuracy in this case.

Table 3 shows the error obtained by varying the number of layers and neurons per layer in each model. The goal is to analyze how the network architecture affects the prediction accuracy. As can be seen from the table, season 3 consistently exhibits lower MSE values across various layer configurations, and season 2 has higher errors, as happened in the previous table. The lowest configuration is obtained with a three-layered transformer with 2 neurons in the first and second layers and 8 neurons in the last one. And again, the transformer trends to outperform LSTM, MLP, and MTNN. MTNN shows consistent performance with lower variability within each row. Finally, the {5, 2, 5} configuration for MLP achieves a lower error in season 1, though there is not a clear improvement in increasing the number of layers. Interestingly, the previous experiment (see Table 2) showed that all three models performed consistently better when using only one layer.

The outcomes of the third experiment are presented in Table 4, explaining the influence of varying the number of epochs. Notably, a discernible trend emerges across most models, wherein an increase in the number of epochs tends to yield poorer results. While certain specific instances display improved outcomes, the predominant pattern suggests a negative impact as the epochs increase. Remarkably, the optimal performance, once again, is attributed to the transformer model, particularly when configured with 400 epochs. Nevertheless, three notable exceptions exist, where the most favorable results are achieved with 100 epochs: MTNN in season 1, LSTM in season 2, and MTNN in season 3.

The final Table 5 illustrates the MSE variations with changes in batch size. Up to a batch size of 8, a discernible improvement is observed across nearly all models and seasons. However, this improvement breaks when changing to a batch size of 10. However, an exception occurs in season 2, where both LSTM and MTNN exhibit enhanced performance with batch sizes of 10 and 6, respectively. It appears that a batch size of 8 stands out as the optimal choice, as it consistently yields favorable outcomes for most models and seasons. Season 3 distinguishes itself with the least variability among the three models, while season 1 displays the highest variability. Throughout, season 2 consistently maintains the highest error rates.

To assess for a visual representation of the evolution of the experiment conducted, we can check Figure 5. The figure depicts the evolution of our experiments, divided into three parts corresponding to each season. These figures show the progression and enhancement of our models' performance. Thus, the figure highlights some discernible improvements and fluctuations in the model's error rates. Thus, it offers a visual understanding of how well the iterative refinement process works throughout the study.

An additional experiment was conducted employing SVR, Table 6, known for its proficiency in regression problems. Despite its reputation, the results from SVR showed a higher MSE compared to the ANN models. Among the different kernel functions tested (radial-basis, linear, polynomial, and sigmoid), the linear kernel exhibited the best performance. Subsequent tests with the regularization parameter (C value) revealed that the models could be improved with optimal parameter tuning, providing MSE values of 781.74, 1514.41, and 655.92 for Seasons 1, 2, and 3, respectively. These findings suggest that while SVR holds promise for regression tasks, the ANN models outperformed SVR in our specific application, highlighting the importance of model selection and parameter optimization.

Table 2. MSE analysis across several neuron configurations.

Neurons	Season 1				Season 2				Season 3			
	MLP	MTNN	LSTM	Transformer	MLP	MTNN	LSTM	Transformer	MLP	MTNN	LSTM	Transformer
2	727.0703	812.8749	648.0278	612.5543	1210.8792	1156.6104	1461.7490	905.1211	633.1957	631.6688	445.8131	403.6778
4	761.5453	812.7620	815.7919	582.4524	1167.7273	1461.6046	1195.8651	873.7854	633.1788	633.2068	433.7654	391.1211
8	651.7265	812.7345	608.7301	474.1287	1461.4097	1461.5884	1199.1933	791.2334	633.1748	633.1878	380.1075	336.5321
16	748.0907	811.8649	625.7525	433.2341	1461.5638	1179.0231	1201.5912	821.3321	633.2013	632.1489	368.7934	312.5677
32	651.9003	812.7378	629.8475	405.1323	1461.6357	1461.6150	1274.4498	811.4523	633.1728	633.2039	601.5950	289.2412
64	651.9463	812.7468	639.9440	412.7656	1273.4418	1461.5961	1187.7895	853.1234	633.4088	633.2004	395.3399	245.6544

Table 3. MSE analysis across several layer configurations.

Layers	Season 1				Season 2				Season 3			
	MLP	MTNN	LSTM	Transformer	MLP	MTNN	LSTM	Transformer	MLP	MTNN	LSTM	Transformer
{4, 12}	660.1096	812.7489	829.5544	673.8134	1461.6023	1461.5814	1191.8903	823.6743	633.2051	633.1778	453.3603	408.7865
{8, 8}	671.7164	812.7472	774.2298	534.9843	1191.2417	1461.6384	1197.9731	801.1234	633.1966	633.1970	372.4158	412.5643
{12, 4}	702.1545	689.8484	812.5779	487.2371	1461.6185	1461.6663	1206.4655	856.8754	633.1993	633.1836	449.3485	375.3423
{4, 4, 4}	812.7486	812.7317	599.4713	456.7862	1461.5789	1461.6094	1207.8741	901.8734	633.1622	633.1984	373.0441	288.1237
{2, 8, 2}	812.7453	812.7363	586.7957	395.7643	1461.6858	1461.5979	1461.6719	932.8712	633.1897	633.1935	376.0373	278.9456
{2, 2, 8}	812.7401	630.9656	812.7364	691.9439	1461.5786	1461.6091	1202.8581	956.5623	633.1951	633.1845	373.4460	248.2332
{8, 2, 2}	812.7320	812.7477	812.7448	703.6512	1461.5914	1461.6101	1461.6394	978.1945	633.2031	633.1999	633.1851	434.5674
{5, 2, 5}	635.1971	812.7465	812.8038	724.0543	1461.6101	1461.6579	1188.7855	897.1432	633.2075	633.1956	633.1572	434.6712

Table 4. MSE analysis across several epoch configurations.

Epochs	Season 1				Season 2				Season 3			
	MLP	MTNN	LSTM	Transformer	MLP	MTNN	LSTM	Transformer	MLP	MTNN	LSTM	Transformer
100	660.1096	630.9656	586.7957	395.7643	1191.2417	1461.6091	1187.7895	801.1234	633.1622	633.1845	372.4158	248.2332
200	642.5931	812.7413	633.9173	413.9488	1461.6623	1461.7607	1217.5633	1027.1459	633.1974	633.2054	359.0523	267.9834
400	812.8193	812.7432	575.2028	405.8734	1219.6749	1461.7106	1501.5736	998.7845	633.1817	633.1931	371.6288	239.1234
600	624.5416	812.7351	594.6237	384.8435	1218.6727	1461.7054	1746.0869	905.3975	633.3051	633.1935	373.8825	297.3412
800	659.6547	812.7394	812.7356	567.8577	1461.6884	1068.4161	1203.4065	886.8734	633.1629	633.2008	425.4872	359.6745
1000	626.7629	812.7319	812.7508	592.8734	1178.8728	1461.7648	1189.0123	802.1432	633.0796	633.1981	418.5740	413.7624

Table 5. MSE analysis across several batch size configurations.

Batch	Season 1				Season 2				Season 3			
	MLP	MTNN	LSTM	Transformer	MLP	MTNN	LSTM	Transformer	MLP	MTNN	LSTM	Transformer
4	634.6581	812.7487	627.3410	596.4537	1228.1390	1461.6653	1489.3365	987.3421	633.2126	633.2084	367.4893	295.6759
6	624.5908	812.7526	812.7392	489.5679	1182.3158	1461.6688	1187.1071	965.8466	633.1971	633.1965	370.3734	279.5634
8	624.5416	630.9656	575.2028	395.7643	1178.8728	1461.6091	1187.7895	801.1234	633.0796	633.1845	359.0523	248.2332
10	812.7350	812.7386	812.7370	478.6945	1461.7332	1461.4590	1230.9931	1032.8745	633.1934	633.1947	386.7438	288.6504

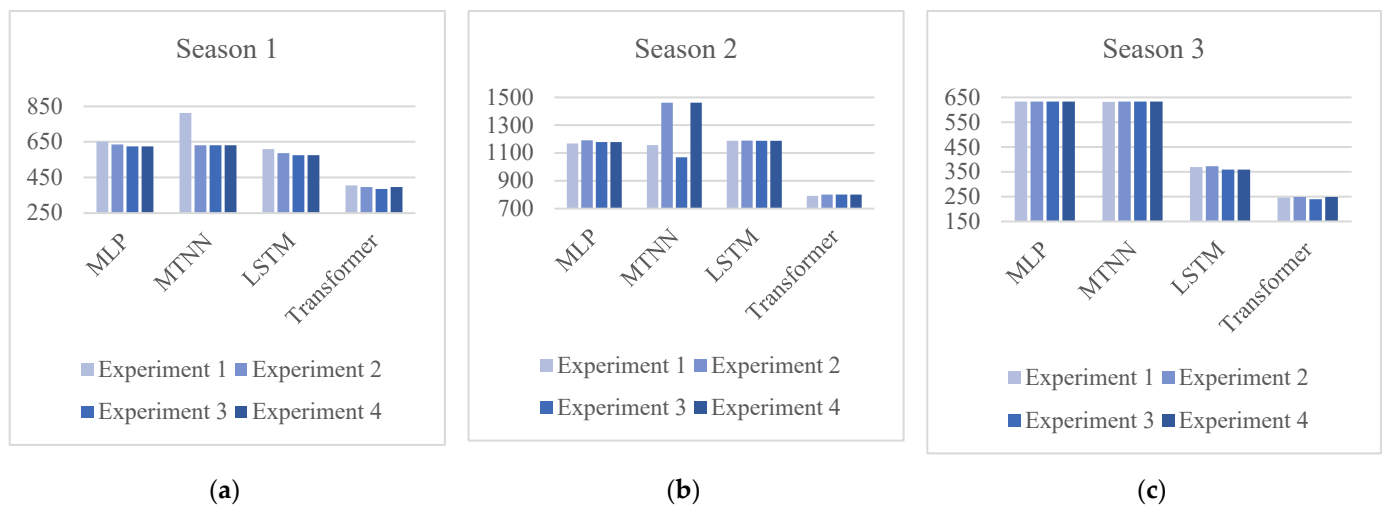


Figure 5. Illustrative evolution of the experimental setup for each season according to each model.

Table 6. MSE analysis across several SVR configurations.

Kernel	Season 1	Season 2	Season 3
Radial-basis	855.56	1672.18	677.61
Linear	808.62	1606.52	644.65
Polynomial	855.65	1672.63	677.72
Sigmoid	855.74	1672.27	677.65

5. Discussion

Our analysis revealed interesting patterns across different seasons. Particularly, season 3 consistently outperformed other seasons, with the transformer model showcasing the lowest error rates. On the other hand, season 2 presented the most significant modeling challenges, as evidenced by its consistently high error rates. This might be due to the increased precipitation and temperature during this season, which provide an optimal environment for pest existence and reproduction. Interestingly, no clear trend emerged across all models regarding improved performance with increasing neurons. The MTNN model stood out in this regard, achieving optimal performance with just 2 neurons in both season 2 and season 3.

The second experiment explored the impact of varying layers and neurons. Similar to the previous experiment, season 3 exhibited consistently lower MSE values across various configurations, while season 2 maintained higher errors. Within this context, a three-layered transformer model with 2, 2, and 8 neurons per layer emerged as the optimal configuration, indicating its superiority over MLP and MTNN models. Notably, MTNN displayed consistent performance with lower variability within each configuration, reinforcing its reliability. This finding, along with the one from the previous experiment, highlights the potential effectiveness of simpler architectures, suggesting that complexity does not always equate to better predictions.

Exploring the influence of epochs and batch size revealed additional knowledge. In general, increasing the number of epochs beyond a certain point led to poorer performance for most models. This finding emphasizes the importance of considering optimal epoch numbers for different models and scenarios to prevent overfitting. Similarly, increasing the batch size up to 8 generally led to improvements, but this effect stopped at a size of 10. Interestingly, season 2 displayed irregular responses, with batch sizes of 10 and 6 offering the best results for LSTM and MTNN, respectively. Additionally, season 3 displayed the least variability in performance across different batch sizes, while season 1 showed the highest. Thus, season 2 continued to demonstrate the highest error rates, suggesting persistent modeling challenges in this specific case.

An experiment was conducted using SVR, which did not demonstrate superior performance compared to other models.

Based on the visual trends observed in the experiment evolution, we can identify some improvements across various architectures. MLP and LSTM models demonstrate positive advances in certain cases. However, the MTNN model only showed a significant improvement from the first to the second experiment. The optimal model was transformer for all three seasons.

6. Conclusions

This study provides a thorough exploration of ANN configurations and hyperparameters, uncovering valuable insights into the modeling process to predict pest numbers in grain crops in Kazakhstan. The experiments with varying neurons revealed the importance of seasonality, with season 3 consistently producing the most favorable outcomes. Note that, in this context, there are distinct periods with different environmental and agricultural conditions throughout the year. These are known as seasons. Our models were designed to consider these variations, which means their performance can differ across seasons. Interestingly, our experiments showed higher MSE in the second season. This suggests a larger amount of varied data during this time. We believe factors such as increased rainfall and favorable conditions for pests during this season might explain these trends. This highlights the impact of seasons on how well our models perform.

Remarkably, the transformer model proved to be a strong performer, displaying the lowest error rates across multiple seasons. The simplicity of architecture proved to be effective, as models with fewer layers demonstrated superior predictions, emphasizing the potential advantages of streamlined structures.

The optimal number of epochs proved to be crucial, as an indiscriminate increase beyond a certain threshold led to diminishing returns for most models. The transformer model got the best performance with 400 epochs. The relationship with batch size showed a general improvement up to a size of 8, beyond which the gains stopped. The exception in season 2, where particular responses were observed, underscores the importance of specific tuning.

Throughout the experiments, the transformer model consistently outperformed LSTM, MLP, and MTNN in terms of predictive accuracy.

For future work, we suggest developing an analytical system to predict pest numbers in grain crops in Kazakhstan. In addition, further exploration and comparison between hyperparameter tuning methods could serve as a promising avenue for future research. While our findings indicate that GridSearchCV is a suitable method for hyperparameter tuning on our specific dataset, it would be valuable to conduct a comparison with other techniques such as evolutionary computation and swarm algorithms. This comparative analysis could shed light on the relative strengths and weaknesses of each method across a wider range of datasets and model architectures.

Finally, we plan to explore the theoretical aspects of developing methods based on ANNs for agricultural applications. While our current paper focuses on practical applications, we recognize the value of exploring theoretical foundations. This exploration would involve investigating algorithmic design, optimization techniques, and mathematical principles leading to ANN architectures.

Author Contributions: Conceptualization, L.G.B.R.; data curation, N.O.; investigation, G.A.; methodology, A.A.; resources, N.O.; software, S.S.; supervision, A.A.; writing—original draft, G.A. and S.S.; writing—review and editing, L.G.B.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ministry of Science and Higher Education of the Republic of Kazakhstan, IRN “AP 19675312 Analytical system for forecasting the dynamics of the number of pests of grain crops in Kazakhstan based on a neural network model”.

Data Availability Statement: The data may be provided free of charge to interested readers by request through the correspondence email.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

ANN	Artificial neural network
CNN	Convolutional neural networks
LSTM	Long short-term memory
ML	Machine learning
MSE	Mean squared error
MTNN	Multi-task neural network
RNN	Recurrent neural network
SVR	Support vector regression

References

- Anderson, J.A.; Ellsworth, P.C.; Faria, J.C.; Head, G.P.; Owen, M.D.; Pilcher, C.D.; Shelton, A.M.; Meissle, M. Genetically engineered crops: Importance of diversified integrated pest management for agricultural sustainability. *Front. Bioeng. Biotechnol.* **2019**, *7*, 24. [[CrossRef](#)]
- Ray, P.P. Internet of things for smart agriculture: Technologies, practices and future direction. *J. Ambient Intell. Smart Environ.* **2017**, *9*, 395–420. [[CrossRef](#)]
- Anastasiou, E.; Fountas, S.; Voulgaraki, M.; Psiroukis, V.; Koutsiaras, M.; Kriezi, O.; Lazarou, E.; Vatsanidou, A.; Fu, L.; Bartolo, F.D.; et al. Precision farming technologies for crop protection: A meta-analysis. *Smart Agric. Technol.* **2023**, *5*, 100323. [[CrossRef](#)]
- Liu, C.; Zhai, Z.; Zhang, R.; Bai, J.; Zhang, M. Field pest monitoring and forecasting system for pest control. *Front. Plant Sci.* **2022**, *13*, 3202. [[CrossRef](#)]
- Prakash, B.; Singh, P.P.; Raghuvanshi, T.S. Chapter 1—Green products in the management of stored food grains: Challenges, recent advances and future prospects. In *Green Products in Food Safety*; Prakash, B., Brillhante de São José, J.F., Eds.; Academic Press: Cambridge, MA, USA, 2023; pp. 1–27. [[CrossRef](#)]
- Adomako, J.; Yeboah, S.; Larbi-Koranteng, S.; Kankam, F.; Oppong-Sekyere, D.; Nboyine, J.A.; Danso, Y.; Osei, M.K.; Oteng-Darko, P. Transgenic plants and its role in insect control. In *New and Future Development in Biopesticide Research: Biotechnological Exploration*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 203–222.
- Wang, D.; Cao, W.; Zhang, F.; Li, Z.; Xu, S.; Wu, X. A review of deep learning in multiscale agricultural sensing. *Remote Sens.* **2022**, *14*, 559. [[CrossRef](#)]
- Hassan, M.; Kowalska, A.; Ashraf, H. Advances in deep learning algorithms for agricultural monitoring and management. *Appl. Res. Artif. Intell. Cloud Comput.* **2023**, *6*, 68–88.
- Attri, I.; Awasthi, L.K.; Sharma, T.P.; Rathee, P. A review of deep learning techniques used in agriculture. *Ecol. Inform.* **2023**, *77*, 102217. [[CrossRef](#)]
- Yan, Y.; Feng, C.-C.; Wan, M.P.-H.; Chang, K.T.-T. Multiple regression and artificial neural network for the prediction of crop pest risks. In *Information Systems for Crisis Response and Management in Mediterranean Countries, Proceedings of the Second International Conference, ISCRAM-med 2015, Tunis, Tunisia, 28–30 October 2015*; Proceedings 2; Springer: Berlin/Heidelberg, Germany, 2015; pp. 73–84.
- Latif, M.S.; Kazmi, R.; Khan, N.; Majeed, R.; Ikram, S.; Ali-Shahid, M.M. Pest prediction in rice using iot and feed forward neural network. *KSII Trans. Internet Inf. Syst. (TIIS)* **2022**, *16*, 133–152.
- Thomas, J.C.S.; Manikandarajan, S.; Subha, T.K. Ai based pest detection and alert system for farmers using iot. In Proceedings of the E3S Web of Conferences, online, 24–25 March 2023; EDP Sciences: Les Ulis, France, 2023; p. 05003.
- Wang, Y.; Huang, P.; Khan, Z.A.; Wei, F. Potential of kazakhstan’s grain export trade. *Ciência Rural.* **2021**, *52*, e20210199. [[CrossRef](#)]
- Akhanova, G.; Nadeem, A.; Kim, J.R.; Azhar, S. A framework of building sustainability assessment system for the commercial buildings in kazakhstan. *Sustainability* **2019**, *11*, 4754. [[CrossRef](#)]
- Petrick, M.; Oshakbaev, D.; Wandel, J. *Kazakhstan’s Wheat, Beef and Dairy Sectors: An Assessment of Their Development Constraints and Recent Policy Responses*; Discussion Paper; Institute of Agricultural Development in Transition Economies (IAMO): Halle, Germany, 2014.
- Ayan, E.; Erbay, H.; Varçın, F. Crop pest classification with a genetic algorithm-based weighted ensemble of deep convolutional neural networks. *Comput. Electron. Agric.* **2020**, *179*, 105809. [[CrossRef](#)]
- Torres, J.F.; Hadjout, D.; Sebaa, A.; Martínez-Álvarez, F.; Troncoso, A. Deep learning for time series forecasting: A survey. *Big Data* **2020**. [[CrossRef](#)] [[PubMed](#)]
- Hossain, M.I.; Jahan, S.; Al Asif, M.R.; Samsuddoha, M.; Ahmed, K. Detecting tomato leaf diseases by image processing through deep convolutional neural networks. *Smart Agric. Technol.* **2023**, *5*, 100301. [[CrossRef](#)]

19. Li, C.; Liu, Y.; Meng, Y.; Duan, Q. Sugarcane juice extraction prediction with physical informed neural networks. *J. Food Eng.* **2024**, *364*, 111774. [[CrossRef](#)]
20. Coulibaly, S.; Kamsu-Foguem, B.; Kamissoko, D.; Traore, D. Explainable deep convolutional neural networks for insect pest recognition. *J. Clean. Prod.* **2022**, *371*, 133638. [[CrossRef](#)]
21. Dong, C.; Zhang, Z.; Yue, J.; Zhou, L. Automatic recognition of strawberry diseases and pests using convolutional neural network. *Smart Agric. Technol.* **2021**, *1*, 100009. [[CrossRef](#)]
22. Khaki, S.; Wang, L.; Archontoulis, S.V. A cnn-rnn framework for crop yield prediction. *Front. Plant Sci.* **2020**, *10*. [[CrossRef](#)] [[PubMed](#)]
23. Xiao, Q.; Li, W.; Kai, Y.; Chen, P.; Zhang, J.; Wang, B. Occurrence prediction of pests and diseases in cotton on the basis of weather factors by long short term memory network. *BMC Bioinform.* **2019**, *20*, 688. [[CrossRef](#)]
24. Kang, J.; Gwak, J. Ensemble of multi-task deep convolutional neural networks using transfer learning for fruit freshness classification. *Multimed. Tools Appl.* **2022**, *81*, 22355–22377. [[CrossRef](#)]
25. Luo, W.; Zhang, C.; Li, Y.; Yan, Y. Mlgnnet: Multi-task learning network with attention-guided mechanism for segmenting agricultural fields. *Remote Sens.* **2023**, *15*, 3934. [[CrossRef](#)]
26. Long, J.; Li, M.; Wang, X. Boundary-aware multi-task neural networks for agricultural field extraction from very high-resolution satellite images. In Proceedings of the 2022 29th International Conference on Geoinformatics, Beijing, China, 15–18 August 2022; pp. 1–5.
27. Zhong, H.; Wang, J.; Jia, H.; Mu, Y.; Lv, S. Vector field-based support vector regression for building energy consumption prediction. *Appl. Energ.* **2019**, *242*, 403–414. [[CrossRef](#)]
28. Chang, H. Multilayer perceptron. In *Machine Learning? A Journey to Deep Learning*; World Scientific: Singapore, 2021. [[CrossRef](#)]
29. Lehna, M.; Scheller, F.; Herwartz, H. Forecasting day-ahead electricity prices: A comparison of time series and neural network models taking external regressors into account. *Energy Econ.* **2022**, *106*, 105742. [[CrossRef](#)]
30. Xiao, Y.; Yin, H.; Zhang, Y.; Qi, H.; Zhang, Y.; Liu, Z. A dual-stage attention-based conv-lstm network for spatio-temporal correlation and multivariate time series prediction. *Int. J. Intell. Syst.* **2021**, *36*, 2036–2057. [[CrossRef](#)]
31. Monteiro, N.R.C.; Oliveira, J.L.; Arrais, J.P. Dtitr: End-to-end drug–target binding affinity prediction with transformers. *Comput. Biol. Med.* **2022**, *147*, 105772. [[CrossRef](#)] [[PubMed](#)]
32. Adnan, M.; Alarood, A.A.S.; Uddin, M.I.; ur Rehman, I. Utilizing grid search cross-validation with adaptive boosting for augmenting performance of machine learning models. *PeerJ Comput. Sci.* **2022**, *8*, e803. [[CrossRef](#)] [[PubMed](#)]
33. Jalali, S.M.J.; Ahmadian, S.; Khosravi, A.; Shafie-khah, M.; Nahavandi, S.; Catalão, J.P.S. A novel evolutionary-based deep convolutional neural network model for intelligent load forecasting. *IEEE Trans. Ind. Inform.* **2021**, *17*, 8243–8253. [[CrossRef](#)]
34. Albuquerque Filho, F.S.d.; Madeiro, F.; Fernandes, S.M.; de Mattos Neto, P.S.; Ferreira, T.A. Time-series forecasting of pollutant concentration levels using particle swarm optimization and artificial neural networks. *Química Nova* **2013**, *36*, 783–789. [[CrossRef](#)]
35. Yang, Y.; Duan, Z. An effective co-evolutionary algorithm based on artificial bee colony and differential evolution for time series predicting optimization. *Complex Intell. Syst.* **2020**, *6*, 299–308. [[CrossRef](#)]
36. Wang, Y. On efficiency properties of an r-square coefficient based on final prediction error. *Stat. Probab. Lett.* **2013**, *83*, 2276–2281. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.