Graph Regression to Determine Hen Productivity

1st Nino P. Sabella
MSc in Data Science
Freie Universität Berlin
Berlin, Germany
a.sabella@fu-berlin.de

2nd Silvia Cambiago
MSc in Data Science
Freie Universität Berlin
Berlin, Germany
s.cambiago@fu-berlin.de

3nd Dogan Samur
BSc in Computer Science
Freie Universität Berlin
Berlin, Germany
d.sam@fu-berlin.de

***Abstract*—This project focuses on addressing a graph regression problem with the objective of predicting daily chicken productivity (egg-laying rates) based on recorded social interactions among chickens housed in a commercial Australian barn. The data used in this study spans 117 non-consecutive days, covering the period from February to December of a single year. For each day, a distinct interaction graph is constructed, where nodes represent individual chickens and edges denote pairwise interactions that occurred between them on that day. These edges are further annotated with rich interaction-level features, including the time of day the interaction took place, the physical location within the barn, and the duration of the interaction.**

To model this data, we employ a Graph Neural Network (GNN) architecture, specifically a Graph Attention Network (GAT), which is well-suited for capturing the structure and dynamics of interaction-based systems. The attention mechanism in the GAT enables the model to learn which interactions are more significant by assigning learned weights to different edges during message passing. This allows the model to focus more heavily on interactions that are potentially more relevant for predicting productivity. After learning node embeddings through the attention layers, these embeddings are aggregated into a single graph-level embedding using a pooling operation. This fixed-size representation of the day's interactions is then passed through a series of fully connected layers to produce a single output: the predicted number of eggs laid on that day.

The model is trained in a supervised fashion using labeled graphs, where each graph is associated with its corresponding day's ground truth productivity value. The training objective is to minimize prediction error using Mean Squared Error (MSE) loss. To assess the model's predictive performance and generalization ability, two standard regression evaluation metrics are used: Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). These metrics provide insight into both the average magnitude and the squared variance of the prediction errors.

This project demonstrates the potential of using GNN-based methods, and specifically attention-driven architectures, to model complex animal interaction patterns in a data-driven way. The approach illustrates how daily behavioral data can be transformed into structured graphs for predictive tasks and highlights the relevance of graph regression techniques for real-world applications in precision agriculture and animal welfare monitoring. Overall, the results support the effectiveness of deep graph learning for forecasting outcomes in social animal systems.

I. Introduction

In 2013, the FAO highlighted in its specialized publication, "Poultry Development Review" [1], that the poultry sector is among the most dynamic and adaptable within livestock farming. In recent years, driven by strong global demand, this sector has seen substantial growth, expanding and becoming more globalized across countries of all income levels.

Just like in every other aspect of society, there has been a constant and massive introduction of advanced technlogy methods to improve the quality and safety of poultry products. For example, IoT-based wearable devices, such as accelerometers and gyroscopes, collect data on animal movements and transmit it to the cloud for in-depth analysis, in order to detect early signs of disease, facilitate a fast intervention and limiting the spread of infections [3].

Another example is the introduction of image processing techniques [2], that offer another cost-effective approach for evaluating poultry behavior, health status, and weight prediction. Computer vision techniques can track movement around feeding areas and detect signs of stress or illness. Thermal cameras are used to check for heat stress, and posture analyses using SVM models have reached high accuracy levels in disease classification [2]. Continuous monitoring enables detection of infections as early as the fourth day, thus enhancing health management and overall animal welfare on farms.

However, while these technologies generate vast amounts of valuable behavioral data, traditional analytic approaches often fail to capture the complex set of social interactions within a flock, and these are factors that can significantly influence egg production outcomes.

To address this, we propose modeling chicken behavior as a dynamic interaction network, where each chicken is represented as a node and their social interactions as edges. This graph-based perspective enables the capture of nuanced behavioral patterns that are critical for understanding and predicting egg-laying outcomes. We leverage a Graph Attention Network, a state-of-the-art graph neural network architecture, which uses attention mechanisms to dynamically weigh the importance of each interaction, better detailed in the Background and Related Work section.

Our project involves transforming detailed daily interaction logs into graphs. The GAT model then processes these graphs to learn complex social dynamics and their impact on productivity, producing predictions that inform more effective hen management. As hen management is difficult to do on short notice, we have not only considered same-day predictions regarding productivity, but also next-day predictions. Our approach offers an entry point towards a robust and scalable solution for optimizing poultry production.

II. BACKGROUND AND RELATED WORK

In the domain of egg production, the application of automation technologies is essential for boosting productivity and quality. In the last decade, computer vision and machine learning have emerged as transformative tools in poultry farming, enabling non-invasive, real-time monitoring and management of chicken health, welfare, and productivity [5]. In fact, machine learning models such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Long Short-Term Memory Networks (LSTMs), and Support Vector Machines (SVMs) have been successfully and widely applied to detect disease symptoms, monitor movement patterns, and identify abnormal behaviors within flocks [4].

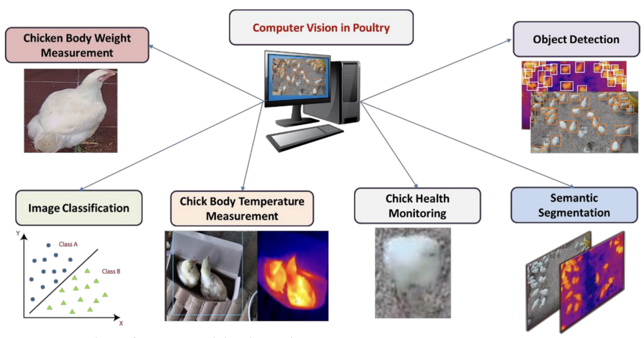

Fig. 1. Applications of computer vision in poultry. [4]

This project builds upon these foundations by employing a Graph Neural Network (GNN), specifically a Graph Attention Network (GAT), to model the complex interactions between chickens within a barn and predict the productivity on the basis of that. A graph is mathematically defined as $G = (V, E)$, where V represents the set of nodes (individual chickens in this project) and E denotes the set of edges (interactions between chickens).

GNNs are a type of deep learning model designed to learn from graph-structured data using message-passing mechanisms, where nodes iteratively aggregate information from their neighbors to update their representations, allowing them to capture the complex relationships in graphs [6].

GNNs typically consist of three primary layers:

- Input Layer: Encodes node features and edge features.
- Hidden Layers: Process data via message passing, where each node's state is updated using a function of its neighbors' states.
- Output Layer: Generates predictions by pooling node-level representations.

The model that will actually be used is the Graph Attention Network (GAT), a GNN variant that incorporate the concept of attention mechanisms to dynamically weight the importance of interactions [7]. This attention mechanism is inspired by the Transformer architecture introduced in "Attention Is All You Need" [8], which showed that attention-based models are highly effective for capturing complex dependencies in structured data.

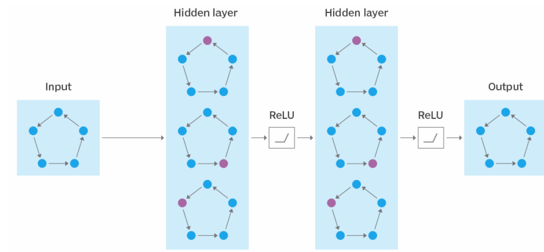

Fig. 2. Structure of a GNN. [11]

In more detail, a GAT works by assigning different weights, or levels of importance, to each connection between nodes during the learning process. Instead of treating all neighboring nodes equally, as in standard GNNs, the GAT computes attention coefficients for each edge, making sure to learn which neighboring nodes are most relevant for predicting a target node's outcome [9] (productivity, in this case). This is particularly advantageous in this context of social animal behavior, where some interactions have a greater influence on productivity than others.

There is currently no literature about applying GATs to chicken interaction or productivity prediction, but there is a GAT-based model, TempoKGAT [10], that has been used on a dataset that tracks the spread of chickenpox through dynamic contact networks. In this context, TempoKGAT has shown significant improvements over traditional GNNs and LSTM-based approaches, especially in capturing the importance of recent and context-specific interactions for predictive tasks. While the ChickenPox dataset is not about poultry, the underlying principle of using temporal graph attention to model and predict outcomes in dynamic social networks is highly relevant.

III. DATASETS

In this project, we had the opportunity to work with two distinct datasets: the Chicken Interaction Dataset and the Productivity Dataset. Together, they enabled us to study whether the social behavior of chickens—captured as a daily interaction graph—could be used to predict barn-level egg productivity.

A. Chicken Interaction Dataset

The first dataset, titled the Chicken Interaction Dataset, recorded fine-grained interactions between chickens inside a commercial Australian barn over the course of one year. In total, it contains 48,245,971 interaction records, each representing one contact between a pair of chickens.

Each interaction is described by the following seven features:

- V1: Hen 1 (unique chicken identifier)
- V2: Hen 2 (unique chicken identifier)
- V3: Contact Duration (in seconds)
- V4: Antenna name, representing the location in the barn where the interaction took place
- time: Unix timestamp indicating when the interaction occurred

- time_: Mean time of the interaction (average between start and end times)
- date: Calendar date of the interaction

The Unix timestamp is a way to track time as a running total of seconds starting from January 1st, 1970, at UTC.

This dataset does not include any target values. Its purpose was to build one graph per day. In each graph, nodes represent individual chickens, and edges represent interactions between them on that day. Each edge could potentially carry attributes like contact duration, time, or location, and we initially considered using these values to enrich the graph.

However, as the project progressed, we discovered that none of these edge features contributed meaningfully to prediction performance. Our final model uses only the structure of the daily interaction graphs. This is essentially about which chickens interacted with each other. In some successful trials, no edge attributes were used. This insight emerged after extensive experimentation and led us to simplify the graph representation while still achieving strong results.

B. Productivity Dataset

The second dataset, referred to as the Productivity Dataset, contained a large number of daily metadata fields describing flock characteristics, environmental conditions, mortality stats, barn features, and egg productivity measurements.

Below is a comprehensive list of columns available in this dataset:

- date
- density
- nodes_n
- crude_clus_coeff
- assortativity_degree
- path
- path_distance
- centr_degree_agg
- centr_degree
- centr_clo
- centr_eigen
- centr_betw
- modularity
- mod_
- size_
- day
- pk
- Farm.ID
- Date
- Year
- Week.day
- Data.source
- Shed.name
- Shed.number
- Flock.ID
- Production.stage
- Breed.name
- Age.of.hens..week.
- Age.of.hens..days.
- Hours.of.light
- Range.access.out
- Range.access.in
- Loose.Birds.caught
- Ammonium.level
- Mortality....Nestbox
- Mortality...Ground
- Mortality.hens...Range
- Mortality.hens...Cull
- Total.mortality
- Mortality....
- Cummulative.mortality....
- Total.feed.consumption.silo.A..kg.
- Total.feed.consumption.Silo.B..kg.
- Feed.consumption.hen.day..g.
- Water.consumption.hen.day..ml.
- Indoor.Minimum.temperature..C.
- Indoor.Maximum.temperature..C.
- Type.of.ventilation
- Clean.eggs
- System.eggs
- Floor.eggs
- Waste.eggs
- Total.eggs
- Number.of.hens
- Laying.rate....
- Total.daily.precipitation..mm..
- Solar.radiation..MJ...meter.squared.per.day.
- Outdoor.Minimum.temperature..C.
- Outdoor.Maximum.temperature..C.
- Vapour.pressure.9am..kilopascal.
- Vapour.pressure.3pm..kilopascal.
- Windspeed..m.s.
- Comments
- flock_date
- feed_per_bird
- water_per_bird
- Relative.humidity.9am
- Relative.humidity.3pm

Although the dataset is extensive, we only used two columns for our final model: Laying.rate...., which served as our regression target, and date, which was used to align each productivity label with its corresponding interaction graph. The laying rate is equal to the number of eggs laid divided by the total number of hens, then multiplied by one-hundred. The rest of the features were not included in training.

$$\text{Lay Rate (\%)} = \left(\frac{\text{Total Eggs Laid}}{\text{Number of Hens}}\right) \times 100$$

To benchmark our model's performance, we also built a simple baseline using a small subset of conventional features from this dataset—specifically, feed_per_bird and water_per_bird. These features are commonly used in poultry science to estimate productivity and represent a more traditional approach.

We trained a standard regression model on these features (without using any graph structure or interaction data) and

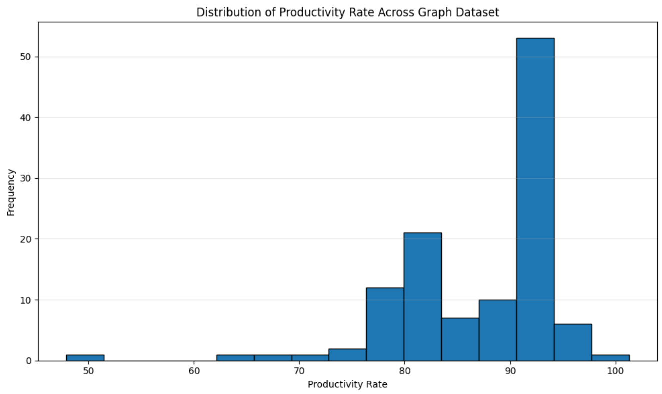

Fig. 3. Productivity Rate Distribution

compared its performance to our graph neural network model. The result was that our graph-based model, using only the interaction structure and no handcrafted features, outperformed the traditional baseline. This suggests that the daily social structure among chickens contains strong predictive signals for productivity—even stronger than commonly used farm-level indicators like feed and water consumption.

C. Summary

In the end, despite access to dozens of environmental, biological, and productivity variables, our final pipeline relied solely on:

- The daily chicken interaction graph (structure only, no edge features), constructed from the Chicken Interaction Dataset
- The daily laying rate, taken from the Productivity Dataset

This framing allowed us to cast the task as a graph-level regression problem, where each graph represents one day of activity, and the model outputs a single continuous value: egg-laying productivity. Our results highlight the surprising power of social behavior patterns for prediction in real-world agricultural settings.

IV. Project Workflow

The project can be divided into three parts. These parts are the exploration of same-day prediction using three GAT model variants, the exploration of one-day-ahead prediction using the same models, and the comparison to different models such as Linear Regression and Random Forest.

We have structured each part into clear sub-sections (e.g., data loading, cleaning, graph conversion, GAT model, etc.). In each sub-section, we describe: the input/output of that step, tools or methods used such as Apache Spark for cleaning and grouping, what decisions we made (how we defined nodes/edges), and any assumptions or challenges. The project workflow section of this paper can be considered as its core aspect.

A. Graph Attention Network Model Process

Same-day prediction refers to grouping one day's worth of data concerning chicken interactions, connecting it to the day's productivity level (measured through productivity rate), and processing it as a graph. The graph is then inserted into the trained Graph Attention Network model and the model gives a prediction about the productivity rate.

Next-day prediction involved the same steps for the most part, with some minor differences. The key difference is that the productivity dates are connected to interactions of the day prior for model input. Next-day prediction is more useful in the real world, as benefactors are likely to be people who need to predict egg lay rate for future dates using current information available to them.

The first step was initially to use Apache Spark to manage the large dataset efficiently. However, it was discovered that the process of moving the data back to pandas from Spark was slower than never moving it to Spark in the first place. Therefore, we decided to no longer use Apache Spark. The official first step ended up simply being to upload the chicken interaction dataset as a csv file and read it using pandas. Columns were renamed to more self explanatory (ie `V1` to `chicken1_id`). We then investigated the data types and ensured that the date was made into a date, and other fields were turned into float or integer formatting after initially being read as objects.

The second step was to do the same procedure for the productivity dataset, from which we only required two columns (productivity rate and date).

The third stage involved traditional cleaning operations including dropping duplicates and dropping rows with NaN values. Each row was technically an observation, so some chicken interactions were dropped. That said, these dropped observations couldn't serve the model as they were missing critical data. After this process, 48,104,016 interactions remained.

The fourth phase was about grouping data by date (day) to get interactions sorted and ready for the transformation into the graph format that our model could read. We did this by making a python dictionary where the key was the date and the value was the filtered dataframe only containing observations from the key's date.

Phase five involved the big conversion from the dictionary to the daily graph. The four parts of this phase were extracting node ID's (chickens), building the edge list and edge features, converting to PyTorch tensors, and incorporating the target label. In the same-day case, the target label was the same day's productivity rate. After doing this for all 117 dataframes in the dictionary, a new graph dataset consisting of 117 graphs was created in Kaggle and plugged into the model notebook for efficiency. For next-day predictions, only 116 graphs were created as one day's interaction data had no matching target label (next closest day's lay rate).

The sixth stage took place in a new model notebook, holding our three GAT models: global mean pooling, global max pooling, and global attention pooling. Global mean pooling is a method that averages the node features across the entire graph, giving equal importance to all nodes and producing a single, fixed-size vector that represents the graph. In contrast, global max pooling selects the maximum value for each feature across all nodes, emphasizing the most activated features and helping the model focus on the strongest signals. Global attention pooling introduces a learnable attention mechanism that assigns different weights to nodes based on their relevance, allowing the model to prioritize more important nodes when forming the graph-level representation. The sixth stage involved padding the graphs, meaning adding zeros (when necessary) to edge and node feature vectors to make all graphs the same size for the model to process. The models were then created, trained, and evaluated using 5-fold cross-validation. The 5-fold cross-validation ensured confidence in our numbers as we were using averages between various model runs as our total evaluation metrics. This reduces the chance of obtaining results that do not match the reality of each model's predictive power.

The seventh and final step was to evaluate the results using critical thinking and applying explainability (XAI) methods to further understand the best performing model's reasoning behind its decisions. This step's outcome will be discussed in the results part of this paper.

B. Linear Regression and Random Forest Process

To establish baseline predictive models for hen egg-laying productivity, we implemented two classical machine learning approaches: Linear Regression and Random Forest Regression. These models were trained on two types of data: daily aggregated behavioral features extracted from chicken interaction data, and environmental features collected independently.

Initially, we imported the chicken interaction dataset, which contains records of contacts between pairs of hens, including the duration of contact, antenna location, and timestamps. To improve readability and consistency, we renamed the columns from their original cryptic names to descriptive ones: `V1` and `V2` to `chicken1_id` and `chicken2_id`, contact duration columns, antenna location, and timestamp columns were similarly renamed.

Dates in the dataset were converted to `datetime` objects to facilitate time-based grouping and filtering. Any rows with missing or invalid dates were removed, ensuring a clean temporal dimension. We then ensured that the chicken identifiers were treated as strings because these are categorical IDs rather than numerical values. To capture unique pair interactions, a combined identifier `both_ids` was created by concatenating `chicken1_id` and `chicken2_id`. Duplicate rows and any remaining missing values were dropped to avoid skewing our summary statistics and model training with incomplete or redundant data.

Since individual interaction events are too granular for predicting daily egg productivity, we aggregated the interaction data by date to derive daily-level behavioral features. The aggregation produced the following metrics:

- Unique Chickens: The number of unique hens appearing as `chicken1_id` and as `chicken2_id` each day.
- Interaction Count: The total number of contact events recorded on that day.
- Contact Duration Statistics: Including the mean, standard deviation, and maximum contact duration among all recorded interactions on that day.
- Unique Pairs: The count of distinct hen pairs interacting, based on the `both_ids` column.
- Unique Locations: The number of unique antenna locations detected, representing different physical zones where interactions occurred.

Additionally, a derived feature called `total_unique_hens` was calculated by summing the number of unique hens appearing in both `chicken1_id` and `chicken2_id` columns, capturing overall daily hen participation.

To quantify how dense the interaction network was each day, we computed `interaction_density` as the ratio of `unique_pairs` to `total_unique_hens`. This metric helps describe the average number of unique contacts per hen.

The productivity target data, containing daily egg-laying metrics, was loaded separately. The key variable of interest was the `Laying.rate....`, representing the daily egg-laying rate as a percentage.

Due to localization issues, decimal points were represented with commas, so these commas were replaced with periods to correctly parse the data as floating-point numbers. The date column was converted to datetime, and only the relevant columns—date and laying rate—were retained.

Behavioral features and productivity targets were merged on the date column, ensuring that each feature vector corresponded to the correct productivity label. An inner join was used to include only days present in both datasets. We selected behavioral features for the first modeling attempt, specifically: interaction count, average contact duration, standard deviation and maximum duration of contact, unique pairs, unique antenna locations, total unique hens, and interaction density.

The dataset was split into training and testing sets using an 80-20 split with a fixed random seed for reproducibility. A Linear Regression model was trained on the training data. This model assumes a linear relationship between features and productivity, making it a simple but interpretable baseline. The model's predictions on the test set were evaluated using:

- Root Mean Squared Error (RMSE), which penalizes larger errors more heavily.
- Mean Absolute Error (MAE), which measures average absolute prediction error.
- Coefficient of determination (R^2), indicating the proportion of variance explained by the model.

The Linear Regression model on behavioral data yielded an RMSE of approximately 5.88, indicating that, on average, the predicted laying rates deviated by this amount from actual values. The R^2 score was moderate, reflecting limited predictive power. Visualizations such as scatter plots of predicted vs. actual values and line plots over time helped confirm the model's trend capture and identify days with larger deviations. To compare, we trained a second Linear Regression model using only environmental features from the productivity dataset. This dataset includes measurements like relative humidity, feed and water per bird, and other environmental and management variables. Non-numerical columns were converted to numeric either by replacing commas with periods and converting to floats or by factorizing categorical variables into integers. Missing values were replaced with zeros to maintain dataset integrity.

Again, an 80-20 train-test split was used, and the model was trained and evaluated similarly. This environmental-only model performed much better than the behavioral models in the linear regression setting, achieving an RMSE of approximately 0.46 and a high R^2 score, indicating a strong linear relationship between environmental factors and laying rate. However, linear models have limited capacity to capture complex nonlinear relationships and interactions between variables. To address this, we applied Random Forest regression, an ensemble method that constructs multiple decision trees and averages their predictions.

Separate Random Forest models were trained on the behavioral and environmental datasets using default hyperparameters and a fixed random seed for reproducibility. The Random Forest model trained on behavioral features showed some improvement over linear regression, with an RMSE of about 5.65 and an R^2 around 0.63, indicating better modeling of nonlinear patterns within the behavioral data. On the environmental features, Random Forest regression performed exceptionally well, achieving an RMSE near 1.21 and an R^2 close to 0.97, reflecting a near-perfect fit on the test data.

These results highlight that environmental variables are strong predictors of egg productivity, which is expected since factors such as feed, water, and humidity directly influence hen physiology.

Behavioral features extracted from interaction data provide additional but weaker signals, likely because social interactions are only one of many factors affecting productivity. While Random Forest models improve upon linear baselines by capturing nonlinearities, they do not fundamentally change which features are most predictive.

Importantly, although these baseline models show that environmental data alone can predict productivity relatively well, they do not leverage the relational and temporal information embedded in the hens' interaction network. Our Graph Attention Network (GAT) model, by contrast, explicitly incorporates the graph structure of hen interactions over time, allowing it to capture complex dependencies and social dynamics that aggregated behavioral or environmental features cannot.

As will be demonstrated in the Results section, the GAT model significantly outperforms both behavioral and environmental baseline models, proving that the combination of graph structure and attention mechanisms is essential to accurately predict egg productivity. Therefore, these baseline models establish important performance benchmarks, but the GAT approach marks a clear advancement by fully exploiting the rich, structured interaction data.

This step-by-step process ensures that data preprocessing, feature engineering, model training, and evaluation are transparent and reproducible, providing a solid foundation for comparing more sophisticated graph-based models.

V. Results

In this section we report and interpret the performance of our different GAT models. We analyze quantitative results such as the RMSE and MAE.

A. Model Evaluation: Egg Productivity Prediction

The baseline productivity rate of the hens had a mean of 87.16 and a standard deviation of 8.34, which provides context for evaluating the predictive accuracy of the models tested. The models were evaluated on both same-day and next-day predictions, using Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) as the primary performance metrics.

1) Same-Day Prediction Results: The simplest approach, using global mean pooling to aggregate node features, resulted in the highest errors. The values were relatively large compared to the variability in the data, indicating that mean pooling fails to capture important differences between nodes in the interaction graph. This is likely due to the oversimplification inherent in averaging all node features equally, which can dilute the influence of key individuals or interactions that are more predictive of egg productivity.

Global max pooling showed a substantial improvement, reducing RMSE to approximately 0.77 and MAE to 0.62. This suggests that focusing on the strongest signals among nodes (via max pooling) helps the model identify more relevant features, but the performance is still not optimal. While max pooling emphasizes the dominant node features, it ignores the contribution of other nodes that may also contain valuable information.

The best performance was achieved with global attention pooling, with RMSE dropping to 0.51 and MAE to 0.32. The attention mechanism enables the model to assign varying weights to different nodes based on their relevance, rather than treating all nodes equally or only focusing on the maximum feature value. This more nuanced aggregation allows the model to capture complex patterns within the graph that correlate strongly with productivity. Importantly, the MAE here is roughly 1/26th of the standard deviation of the productivity data, highlighting that the model predictions are very close to observed values relative to natural variability.

2) Next-Day Prediction Results: When predicting egg productivity for the following day, the same trends were observed but with generally lower error magnitudes. Mean pooling again showed the highest error, with RMSE near 0.89 and MAE around 0.72. Max pooling provided slight improvements with RMSE and MAE dropping to about 0.85 and 0.69 respectively, still indicating inadequate modeling of the graph information.

Attention pooling continued to outperform both alternatives, with RMSE falling to 0.48 and MAE to 0.36. This further supports the effectiveness of the attention mechanism in capturing

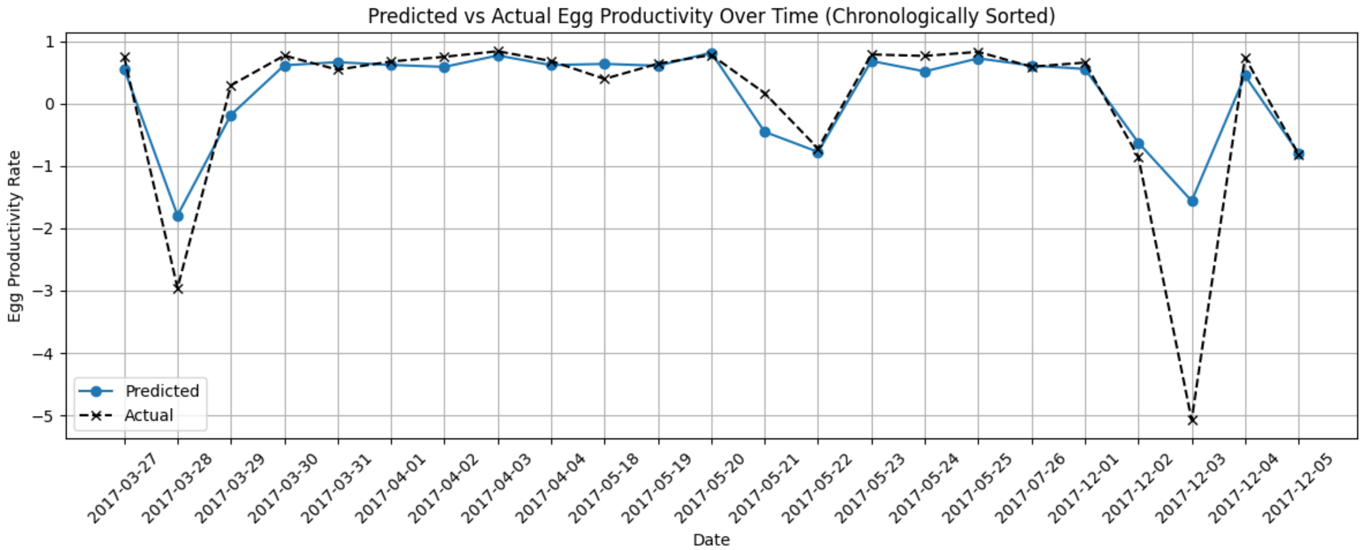

Fig. 4. Same-Day Global Attention Performance

relevant relational information that influences productivity on subsequent days. The consistent advantage of attention pooling across both same-day and next-day predictions demonstrates the robustness and stability of this approach in modeling temporal dependencies and node interactions.

3) Comparison to Environmental Baselines: It is important to highlight that these graph-based results significantly surpass the predictive capabilities of simpler environmental models. Linear Regression using environmental factors alone achieved an RMSE around 0.46, which is comparable only to the attention pooling GAT results but lacks the capacity to capture complex nonlinear relationships or interactions between individuals.

Random Forest models applied separately to behavioral and environmental features showed mixed results. Behavioral data Random Forests improved somewhat over linear models (RMSE 5.65), but still lagged far behind graph attention models. Environmental feature Random Forests performed very well (RMSE 1.21, R^2 near 0.97), indicating environmental conditions like feed, water, and humidity are strong predictors. However, these models do not integrate social interaction patterns, which are essential for capturing the full complexity of egg productivity dynamics.

The graph attention network surpasses both Linear Regression and Random Forest baselines by leveraging relational data in the form of interaction graphs, thereby incorporating social and temporal structures that simple aggregated features cannot represent. This demonstrates that while environmental variables provide important baseline signals, the graph-based approach adds valuable context, leading to improved prediction accuracy.

B. Explainable AI (XAI) Analysis

To better understand the contributions of different features, especially edge attributes, we performed permutation-based feature importance analysis. This method involves randomly shuffling each edge feature to measure the resulting increase or decrease in prediction error, which indicates the feature's importance to the model.

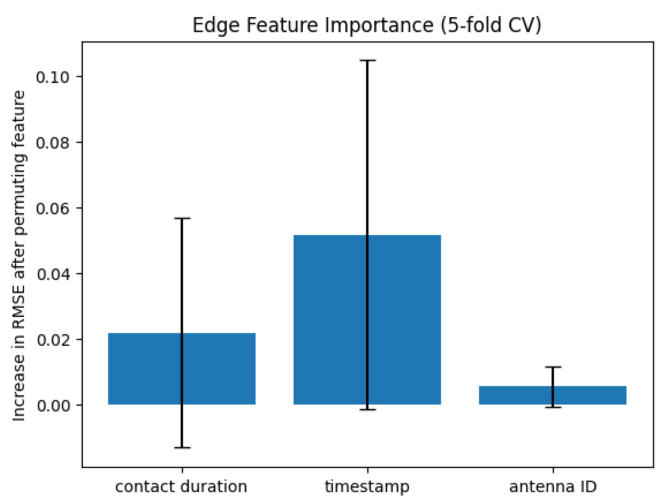

Fig. 5. Next-Day Edge Feature importance

1) Next-Day Edge Feature Importance: For next-day predictions, permuting contact duration led to a negligible increase in RMSE of approximately 0.02, timestamp resulted in a slightly higher increase of 0.06, and antenna ID caused an increase of about 0.01. The small magnitudes of these changes suggest that the model depends only minimally on these edge attributes. Timestamp, while having the highest relative importance among them, still contributes very little compared to node features and the graph structure itself.

This implies that the model mainly utilizes the topology of the graph — how hens are connected and their individual features — rather than detailed temporal or spatial attributes of interactions. The weak reliance on edge attributes indicates that the complex relationships encoded in node features and connectivity patterns are sufficient to explain productivity variations.

2) Same-Day Edge Feature Importance: For same-day data, a different pattern emerged. Shuffling contact duration and timestamp actually decreased RMSE by 0.45 and 0.60 respectively, suggesting that these features may introduce noise or misleading signals, reducing model performance when included as-is. Antenna ID caused a slight increase in RMSE (0.01) when permuted, indicating a minor positive contribution but overall very limited importance.

This counterintuitive result highlights that some edge attributes may be noisy or irrelevant in same-day predictions, and that their removal or randomization can improve model robustness. It further supports the notion that the model's predictive power primarily derives from node features and connectivity, rather than fine-grained edge details.

C. Model Performance Without Edge Attributes (Same-Day)

To verify the limited role of edge attributes, we retrained models excluding all edge features. Results showed:

- Mean pooling RMSE decreased to 0.54, MAE to 0.32, showing slight improvement but still relatively poor performance. - Max pooling had RMSE of 0.52 and MAE of 0.38, slightly better in RMSE but slightly worse in MAE compared to mean pooling. - Attention pooling improved to an RMSE of 0.46 and MAE of 0.31, representing the best performance.

The attention pooling model without edge attributes achieved an MAE approximately 1/27th of the standard deviation, reinforcing its strong predictive capacity relative to data variability. These results demonstrate that the removal of edge attributes does not harm, and can sometimes improve, predictive performance. It confirms that the GAT model primarily exploits node-level information and graph topology rather than relying on edge feature detail.

D. Summary and Implications

Overall, the results demonstrate that the graph attention pooling mechanism is essential for effective egg productivity prediction in this domain. By weighting node contributions intelligently and incorporating the complex relational structure among hens, the model captures nuances that are lost in simpler pooling or baseline environmental models.

The minimal importance and potential negative impact of edge attributes suggest that future modeling efforts could focus on enhancing node feature representations and graph connectivity rather than detailed edge annotations. Additionally, the superiority of GAT models over linear and Random Forest baselines underscores the value of leveraging graph-based approaches to integrate social behavior and environmental context.

These insights provide a strong foundation for future work aiming to optimize productivity prediction through advanced graph neural network architectures and refined feature engineering focused on nodes and network structure.

VI. Conclusion and Future Work

In this study, we developed and evaluated Graph Attention Network (GAT) models to predict egg productivity in hens based on their social interaction graphs. Our results demonstrated that GATs significantly outperform traditional environmental models such as Linear Regression and Random Forests, highlighting the importance of capturing complex relational and behavioral patterns that are not accessible through static environmental features alone. The attention mechanism proved especially effective in identifying and weighting the most relevant nodes within the interaction graphs, leading to consistently lower prediction errors.

This work underscores the strong potential of graph-based deep learning methods for modeling animal behavior and productivity, offering valuable insights for precision farm management. By accurately predicting productivity from behavioral data, these models can contribute to more informed decision-making, optimizing conditions that promote animal welfare and farm efficiency.

For future research, several promising directions emerge. First, incorporating more advanced architectures, such as temporal graph neural networks, could better capture dynamic changes in behavior over longer periods. Second, integrating environmental variables directly with graph data may enhance model robustness and interpretability by combining physiological and social factors. Third, further exploration of edge features and their role could refine model inputs, especially given the low importance observed in this study.

Additionally, extending data collection to cover longer time spans and different farm conditions would improve generalizability. Finally, from a technical perspective, experimenting with different pooling mechanisms, graph augmentations, or hybrid models may reveal further improvements.

Overall, this study lays a strong foundation for applying graph neural networks to animal behavior analysis, opening pathways for smarter, data-driven farm management practices.

VII. Bibliography

This section lists the references cited in the paper. We include key foundational papers on graph attention networks, relevant works on graph machine learning frameworks, and recent studies related to poultry monitoring and behavior analysis.

References

[1] O. Thieme, "Poultry Development Review," FAO.

[2] C. Okinda et al., "A review on computer vision systems in monitoring of poultry: A welfare perspective," *Artificial Intelligence in Agriculture*, vol. 4, pp. 184–208, 2020, doi: 10.1016/j.aiia.2020.09.002.

[3] M. M. Ahmed, E. E. Hassanien, and A. E. Hassanien, "A smart IoT-based monitoring system in poultry farms using chicken behavioural analysis," *Internet of Things*, vol. 25, p. 101010, Apr. 2024, doi: 10.1016/j.iot.2023.101010.

[4] E. Supriyanto, R. R. Isnanto, and S. Hadi Purnomo, "Computer Vision in Chicken Monitoring System Using Machine Learning: A General Review," *E3S Web of Conferences*, vol. 448, p. 02014, 2023, doi: 10.1051/e3sconf/202344802014.

[5] Z. Wu, H. Zhang, and C. Fang, “Research on machine vision online monitoring system for egg production and quality in cage environment,” *Poultry Science*, vol. 104, no. 1, p. 104552, Jan. 2025, doi: 10.1016/j.psj.2024.104552.

[6] B. Khemani, S. Patil, K. Kotecha, and S. Tanwar, “A review of graph neural networks: concepts, architectures, techniques, challenges, datasets, applications, and future directions,” *Journal of Big Data*, vol. 11, no. 1, p. 18, Jan. 2024, doi: 10.1186/s40537-023-00876-4.

[7] J. Zhou et al., “Graph neural networks: A review of methods and applications,” *AI Open*, vol. 1, pp. 57–81, 2020, doi: 10.1016/j.aiopen.2021.01.001.

[8] A. Vaswani et al., “Attention is All You Need,” *Advances in Neural Information Processing Systems*, vol. 30, pp. 5998–6008, 2017.

[9] A. G. Vrahatis, K. Lazaros, and S. Kotsiantis, “Graph Attention Networks: A Comprehensive Review of Methods and Applications,” *Future Internet*, vol. 16, no. 9, p. 318, Sep. 2024, doi: 10.3390/fi16090318.

[10] L. Sasal, D. Busby, and A. Hadid, “TempoKGAT: A Novel Graph Attention Network Approach for Temporal Graph Analysis,” arXiv preprint arXiv:2408.16391, Dec. 2024, doi: 10.48550/arXiv.2408.16391.

[11] TechTarget Contributor, “Graph neural networks (GNNs),” *TechTarget*, 2023. [Online]. Available: https://www.techtarget.com/searchenterpriseai/definition/graph-neural-networks-GNNs.