AI-Driven Conjecture Formulation in Mathematics NSF AIMing (Final Report)

Farhatul Janan and Morolake Fatunmbi

1. Introduction

A mathematical conjecture is a statement or mathematical proposition that is correct based on patterns, observation, or instincts but not proven or disproven. These observations or patterns can arise from experimental results, which can serve as hypotheses to test the pattern (Nickerson, 2009). An example of a mathematical conjecture is the Goldbach Conjecture, which posits that every even integer greater than two can be expressed as the sum of prime numbers (for example, 7+2+11 = 20 (even number)) (Nickerson, 2009).

Kosova et al. (2023) highlighted several math conjectures that have been carried out to derive or disprove mathematical conjectures. They acknowledge that recent advancements in computer programming have greatly expanded the ability to explore and test mathematical conjectures.

With its robust libraries and computational power, Python has become an essential part of this process (Hill, 2016). It has allowed us to test for conjectures across large datasets, discover complex patterns and relationships, the validation of conjectures through extensive analysis, and identify counterexamples that can disprove them (Wang et al., 2023).

This project uses graph theory as a data source to uncover patterns and relationships within network flow and characteristics. Graph theory is a branch of mathematics that uses graphs to study relationships between objects (Mondal & De, 2017). A graph typically consists of vertices (nodes) and edges (links) that connect to model pairwise relationships between objects (Ahuja et al., 1993). Graph theory can be used in network analysis to study social networks, power grids, communication lines, etc., or in computer science for routing and network optimization. Graph theory can also be used to solve optimization in operational research to find the shortest path, maximum flow, etc.

Several features of graph theory can offer discoveries or patterns to formulate hypotheses for mathematical conjecture. One such feature is network flow and its relationship with network structure. In network flow, the objective is to move flow through a network with a specific goal depending on the problem type. This could involve maximizing flow, finding the shortest path, or minimizing the cost of flow, among other objectives.

Networks can represent systems such as water pipelines, road traffic, supply chains, or data networks, where vertices denote locations (real or virtual) and edges represent the paths that connect these points (flow paths or routes). Each edge has a flow capacity that limits the flow that can pass through it, introducing constraints within the network.

Applying machine learning algorithms, graph theory's structure and flow properties can help reveal patterns and relationships that can lead to the formulation of mathematical conjectures that can help specifically in network flow structure and help optimize flow-related problems.

1.1 Network flow questions to address

Max-flow problem: This involves maximizing flow from a source node (start point) to a sink node (endpoint) while within the flow capacity.

Min-cut problem: This refers to the smallest set of edges that when removed, could disconnect the source from the sink.

Based on the work of Wood (1993), the max flow of a network is equal to the min-cut of the network. We will apply this approach to find the maximum flow in the network.

2. Method

This section provides a brief overview of our datasets, and the process used to create them. Additionally, it includes a concise introduction to the baseline models and alternate models utilized in this project. Finally, we describe the training and evaluation paradigm employed in this work.

2.1 Creation of Data

We construct a three-level directed graph as follows:

- Level 1: We randomly select the number of nodes from set $S_1 = \{1,2,3,4,5,6\}$
- Levels 2 and 3: We randomly select the number of nodes from set $S_2 = \{4,6,8,10,12,14,16\}$

After selecting nodes for each level, we initially generate all possible arcs from Level 1 to Level 2 and from Level 2 to Level 3. Then, we randomly select 50% of these arcs in each case, allowing us to create unique graphs with the same node structure. As we are selecting 50% of arcs, that's why we ensure that all elements in set S_2 remain even numbers.

To assign capacities, we add a fixed capacity at each node within a level, ensuring the total capacity across each level is 10. Since we are primarily interested in node capacity as it impacts maximum flow, each arc is assigned a large capacity (10 in this case).

For predictor variables, we use eight key features:

- Number of nodes in Level 1, Level 2, and Level 3
- Number of arcs from Level 1 to Level 2 and from Level 2 to Level 3
- Number of overlapping arcs from Level 1 to Level 2 and from Level 2 to Level 3 (overlapping arcs are defined as outgoing arcs from two nodes that converge on the same node in the next level)
- Number of arcs in the min-cut, where the min-cut's capacity represents the graph's maximum flow

Our outcome variable is the maximum flow within each directed graph. To find the maximum flow, we connect a source node s to the level-1 nodes and sink node t to all level-3 nodes. We then find the maximum flow from s to t. Following this methodology, we generate 1,799 unique graphs, providing a sample size of 1,799 data points for this study.

2.2 Baseline Model

We apply two regression models: a **multiple linear regression model** and a **random forest (RF) regression model**. The linear regression model assumes a linear relationship between predictors and the outcome, providing straightforward interpretation by showing how much the outcome changes with each unit change in a predictor. This model is a good starting point to check the existence and strength of any linear relationship and to evaluate initial model performance.

We then extend our analysis with an RF model. Unlike linear regression, random forests capture complex, non-linear relationships within the data. Additionally, random forests have the robustness to overfitting, outliers and noise. This approach allows us to explore both linear and non-linear patterns in the data and to compare the performance and insights from the two models.

2.3 Alternate model

We perform **multicollinearity analysis** to find the correlations between the predictors with the help of correlation matrix and variance inflation numbers. We then also introduce non-additive association of predictors to check whether the performance of multiple linear regression models improves. For the RF model, we perform model tuning to search for hyperparameters that improve the model performance. We also apply a support **vector regression** (SVR) model to capture the non-linear relationship compared to RF model. This also allows us to see if the SVR outperforms the RF model as SVR works well with small to medium datasets, and it also handles model complexity well through hyperparameters like the kernel type, regularization parameter (C), and epsilon (ε). We use the rbf kernel, C as 1.0, epsilon as 0.1 in SVR.

2.4 Training and evaluation paradigm

We divide the data into an 80-20 rule where we randomly select 80% data for training purposes and 20% for testing the models. To mitigate the risk of overfitting, we perform cross-validation for each of the models. This also provides us with a more reliable estimate of model's performances by using multiple train-test splits. Specifically, we employ 10-fold cross-validation in this project to balance between bias and variance. For multiple regression models (baseline model), we aim to improve our model performance by checking the multicollinearity between predictors and non-additive associations of predictors. Similarly, for RF mode, we tune the parameters to check whether the model performance improves.

3. Results

3.1 Baseline Model

Multiple Linear Regression

We apply sklearn for linear regression. As we mentioned, we randomly divide data into training (80%) and test (20%) datasets. We get the following results for the coefficients in Table 1.

Table 1: Coefficients' values

Sl No.	Coefficient Name	Value

1.	Fixed coefficient	5.6351
2.	Number of nodes in level 1 (V1)	0.5940
3.	Number of nodes in level 2 (V2)	-0.2138
4.	Number of nodes in level 3 (V3)	0.0325
5.	Number of arcs from level 1 to level 2 (V4)	0.3897
6.	Number of arcs from level 2 to level 3 (V5)	-0.0886
7.	Number of overlapping arcs from level 1 to level 2 (V6)	-0.3137
8.	Number of overlapping arcs from level 2 to level 3 (V7)	0.0884
9.	Number of arcs in min-cut (V8)	0.0246

Here, we observe that majority contribution the max flow is due to the constant term, β_0. Among the predictors, V1, V2, V4, and V6 have higher contributions to the outcome compared to the other predictors. We again run another linear regression model with these four predictors and add their coefficients' values in Table 2.

Table 2: Coefficients' values (with four predictors)

Sl No.	Coefficient Name	Value
1.	Fixed coefficient	5.6646
2.	Number of nodes in level 1 (V1)	0.6291
3.	Number of nodes in level 2 (V2)	-0.1745
5.	Number of arcs from level 1 to level 2 (V4)	0.3857
7.	Number of overlapping arcs from level 1 to level 2 (V6)	-0.3149

In Table 2, we can see a slight change in V1 and V2, and the other coefficients' values remain almost the same. We also add Figure 1 and Figure 2 that shows the comparison of actual value with the test value by the residuals (Test Value- Actual Value) for both runs respectively.

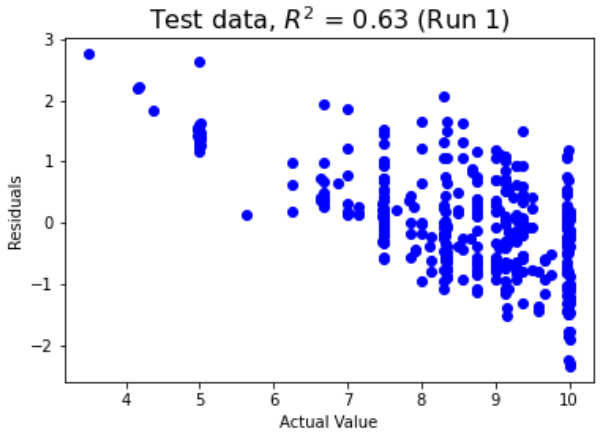

Figure 1: R^2 value (Run 1)

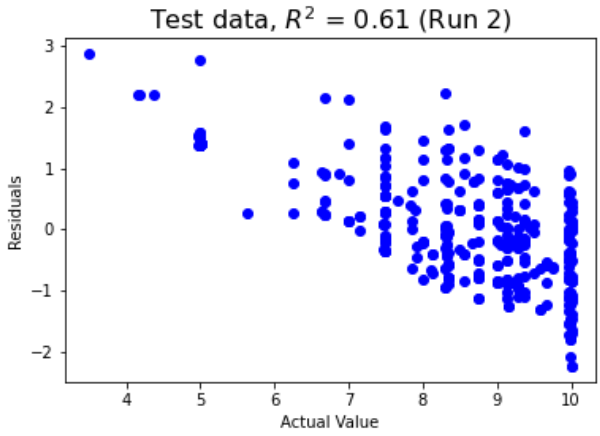

Figure 2: R^2 value (Run 2)

Random Forests (RF)

We then use the RF regression model to predict the max flow based on the predictors. In this model, we find R^2=0.73, and mean-squared error (MSE)=0.71. Table 3 shows a comparison between actual and predicted values using a sample.

Table 3: Sample of actual vs predicted max flow values

Datapoint	Actual	Predicted
265	8.34	7.372838
360	5.00	4.811063
58	8.35	8.442067
284	8.75	8.651700
577	6.67	8.033428

3.2 Alternate Model

Multicollinearity and Non-additive Association of Predictors

First, we check the multicollinearity among our predictors. Figure 3 demonstrates the correlation matrix. Based on the threshold set by Kim (2019), we can see there exists a strong correlation between V4 and V6 (0.98), and V5 and V7 (1.00). We also check the variance inflation number (VIF) for each of the variables, they also indicate the same thing. We aim to delete one of correlated predictors from each pair of correlated predictors. With that, we run for four cases. Table 4 records the mean squared error (MSE), mean R^2 value, and standard deviation of R^2 values.

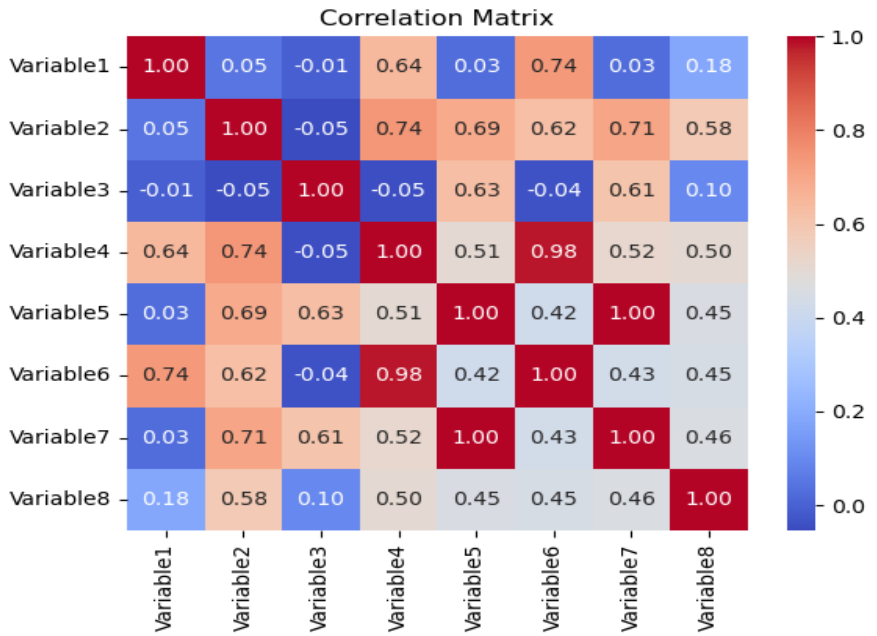

Figure 3: Correlation Matrix

After that we try with the non-additive association of predictors to check if our model improves based on that. We try all combinations of pairwise associations and add notable results in Table 5.

Table 4: Results after removing predictors due to multicollinearity

S/N	Cases	MSE	Mean R^2 values	Standard deviation of R^2 values
1.	Removing V6 and V7	0.97709	0.50	0.09
2.	Removing V4 and V5	0.93975	0.51	0.09
3.	Removing V6 and V5	0.97578	0.50	0.09
4.	Removing V4 and V7	0.9413	0.51	0.09

Table 5: Results with the addition of new predictors capturing non-additive association

S/N	Addition of new predictors based on the association of existing predictors	MSE	Mean R^2 values	Standard deviation of R^2 values
1.	V4*V6	0.6452	0.67	0.07

2.	V4*V6, V5*V7	0.641	0.67	0.07
3.	V4*V6, V3*V5	0.6378	0.67	0.06
4.	V4*V6, V4*V8, V3*V5	0.6066	0.68	0.06
5.	V4*V6, V4*V8, V3*V5, V5*V6	0.598	0.69	0.06

Finding Significant Factors and Tuning of Random Forest (RF)

We use a random forest regressor to identify the key features influencing the maximum flow and then try to derive a relationship linking these features. After training our dataset using random forest, we further carry out feature importance shown in Table 6 i.e. identify the most influential feature for predicting maximum flow.

Table 6: Random forest feature importance table

S/N	Features	Importance
1	V8	0.883
2	V4	0.029
3	V7	0.023
4	V5	0.020
5	V6	0.018
6	V1	0.016
7	V3	0.010
8	V2	0.0008

We performed model optimization using Grid Search to improve performance by exploring a range of parameter values. This approach allowed us to test and balance the trade-off between underfitting and overfitting. Table 7 records the range of parameters that we apply for tuning and the best parameter that we find in this work.

Table 7: Range of hyperparameter and the best parameter

S/N	Hyperparameters	Tested optimization	Best Parameter
1	Number of estimators (trees)	50, 100, 200	100
2	Maximum tree depth	None,10, 20, 30	None
3	Minimum sample per split	2, 5, 10	10
4	Minimum samples leaf	1, 2, 4	4
5	Maximum features	square root, log2, None	None
6	Bootstraps	True, False	True

Support Vector Regression (SVR)

We also apply SVR to predict the network flow in our problem. SVR has the flexibility to capture both linear and non-linear relationships along with its robustness to outliers and it also reduces overfitting using a regularization term. Applying rbf kernel, C as 1.0, epsilon as 0.1 in SVR, we get R^2= 0.86 and MSE=0.26. Figure 4 represents the residual graph from the SVR model.

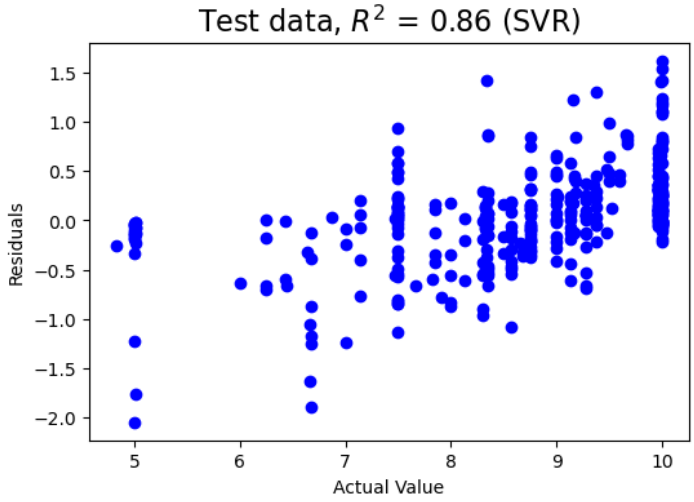

Figure 4: Residual errors and R^2 value in SVR

4. Discussion

4.1 Baseline Model

Multiple Linear Regression

Now, let's look at the model performances based on error rate. As our outcome is a continuous variable, we choose MSE to estimate the error rate. Let's look at the linear regression model first. For the first run with all the predictors, we get MSE= 0.778 while for the second run with significant predictors, we get MSE= 0.813. Based on MSE we can conclude that Run 1 considering all the predictors perform better in case of multiple linear regression. However, we can't draw any conclusions based on R^2 as they may indicate overfitting.

We have also performed 10-fold cross validation to find the mean and standard deviation of R^2 value for both runs. Table 8 records them. This also shows that Run 1 performs better than Run 2.

Table 8: Mean and Standard Deviation of R^2 using 10-fold cross-validation

Run 1	**Run 2**
Mean R^2 values= 0.59	Mean R^2 values= 0.58
Standard Deviation=0.08	Standard Deviation=0.08

Random Forests

The R^2 value of 0.73 indicates that only 73% of the variability in the target variable is explained by the features used in the model. This R^2 value suggests that the model achieved a fairly good performance, an indication of strong correlation between our predictor variables and our outcome (maximum flow). Additionally, MSE of 0.71 indicates that the squared difference between predicted and actual maximum flow values is small. Reasonable as indicated by R2 value. Altogether this evaluation of our random forest model suggests our model can accurately capture most patterns within the dataset.

Comparing our actual values with the predicted values, the model's predictions are a bit closer to the actual values. The predicted values are generally within a small range of the actual values, which reflects a low prediction error and adds further evidence of the model's accuracy. An example is sample 58, the predicted value of 8.44 is close to the actual value of 8.35, same thing can be seen in sample 284 where the predicted value of 8.65 is also very close to the actual value of 8.75. The largest discrepancy in this sample is for Sample 577, where the actual value is 6.67, and the model predicted 8.03. While this indicates the model's limitations for certain instances, most predictions in this sample are reasonably accurate.

The relatively high R-squared suggests that the selected Predictor features are relevant to predicting the outcome feature (maximum flow) values. Identifying the features that contribute most to predictions can enhance interpretability and potentially allow further feature engineering to refine the model.

4.2 Alternate Models

Multicollinearity and Non-additive Association of Predictors

It is noticeable in Table 4 that the MSE increases, and R^2 decreases in each case compared to cases where we considered all the variables in multiple linear regression models of base model. This indicates that the redundant information these variables may have still has some predictable power in our model. On another note, when we are removing predictors, we are also increasing bias in the model. So, that may also be the reason for deteriorating performances in these cases. From Table 5, we observe that the result both in terms of R^2 and MSE improves significantly due to the addition of association between predictors compared to our result in Table 4 and multiple linear regression model of base model. Still, we can't say with certainty as this may also be due to overfitting.

Finding Significant Factors and Tuning of Random Forest (RF)

Our results from Table 6 show that the number of arcs in min-cut (V8) is the most important feature. We then reduce the features (threshold of importance to 0.02) to minimize feature variables and carry out a second run on our model. Our results showed an R^2 of 0.66 and MSE of 0.92 indicating that some of the features excluded from the model, although minimal, had an influence on our model performance.

With the best parameter from the model tuning, we run RF model again and find R^2 as 0.65 and MSE as 0.92 indicating that the tuned model did not improve the predictive performance of the model comparing both MSE and R^2. If our sample size was higher, maybe it would have improved.

Support Vector Regression (SVR)

Based on both R^2 and MSE, SVR seems to outperform all the cases, (improved versions of linear regression and RFR models). As R^2 Sometimes indicating overfitting of the model, we perform 10-fold cross-validation. We get mean R^2= 0.82, and standard deviation of R^2 values = 0.03. Thus, SVR captures the important patterns in our data compared to the other models that we applied here. We also know that SVR is designed to focus on data considering noise and outliers, thus the risk of overfitting is less in this model.

5. Conclusion

In this work, it seems that SVR works best compared to other models that we applied. We think the sample size is the main reason behind that. SVR works well with small to medium datasets and can handle complex non-linear relationships due to the kernel functions. On the other hand, RFR works best when we have a large dataset with enough diverse trees to provide more accurate predictions. We have used a dataset with 1799 data points (dataset size is medium), so SVR works better than RF. On another note, RF indicates the number of arcs in the min-cut (V8) as the most important factor which is also analogous to the theories of network flow. As the maximum flow in a network is equal to the capacities of arcs in the min-cut, the number of arcs in the min-cut has a significant impact on the network flow especially due to the addition of some fixed capacities to the nodes in each level. Thus, the result from our work is consistent with the theory.

References

1. Ahuja, K. R., Magnanti, L. T., & Orlin, B. James. (1993). Network Flows: Theory, Algorithms, and Application. In *Library of Congress Cataloging-in-publication Data*.
2. Hill, C. (2016). *Learning Scientific Programming with Python*. Cambridge University Press. https://doi.org/DOI: 10.1017/CBO9781139871754
3. Kim, J.H., 2019. Multicollinearity and misleading statistical results. *Korean journal of anesthesiology*, *72*(6), pp.558-569.
4. Kosova, R., Kapçiu, R., Hajrulla, S., & Kosova, A. M. (2023). A Revıew of Mathematical Conjectures: Exploring Engaging Topics for University Mathematics Students. *International Journal of Advanced Natural Sciences and Engineering Researches*, 180–186. https://doi.org/10.59287/as-ijanser.581
5. Mondal, B., & De, K. (2017). An Overview Applications of Graph Theory in Real Field. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology © 2017 IJSRCSEIT*, *5*(2), 2456–3307. www.ijsrcseit.com
6. Nickerson, S. R. (2009). Mathematical Reasoning: Patterns, Problems, Conjectures, and Proofs. In *Journal GEEJ* (1st ed., Vol. 7, Issue 2). Psychology Press. https://doi.org/10.4324/9780203848029

7. Wang, H., Fu, T., Du, Y., Gao, W., Huang, K., Liu, Z., Chandak, P., Liu, S., Van Katwyk, P., Deac, A., Anandkumar, A., Bergen, K., Gomes, C. P., Ho, S., Kohli, P., Lasenby, J., Leskovec, J., Liu, T.-Y., Manrai, A., … Zitnik, M. (2023). Scientific discovery in the age of artificial intelligence. *Nature*, *620*(7972), 47–60. https://doi.org/10.1038/s41586-023-06221-2