# Information fusion-based genetic algorithm with long short-term memory for stock price and trend prediction

**Genetic**
**Algorithms**
**Experiments and Results**

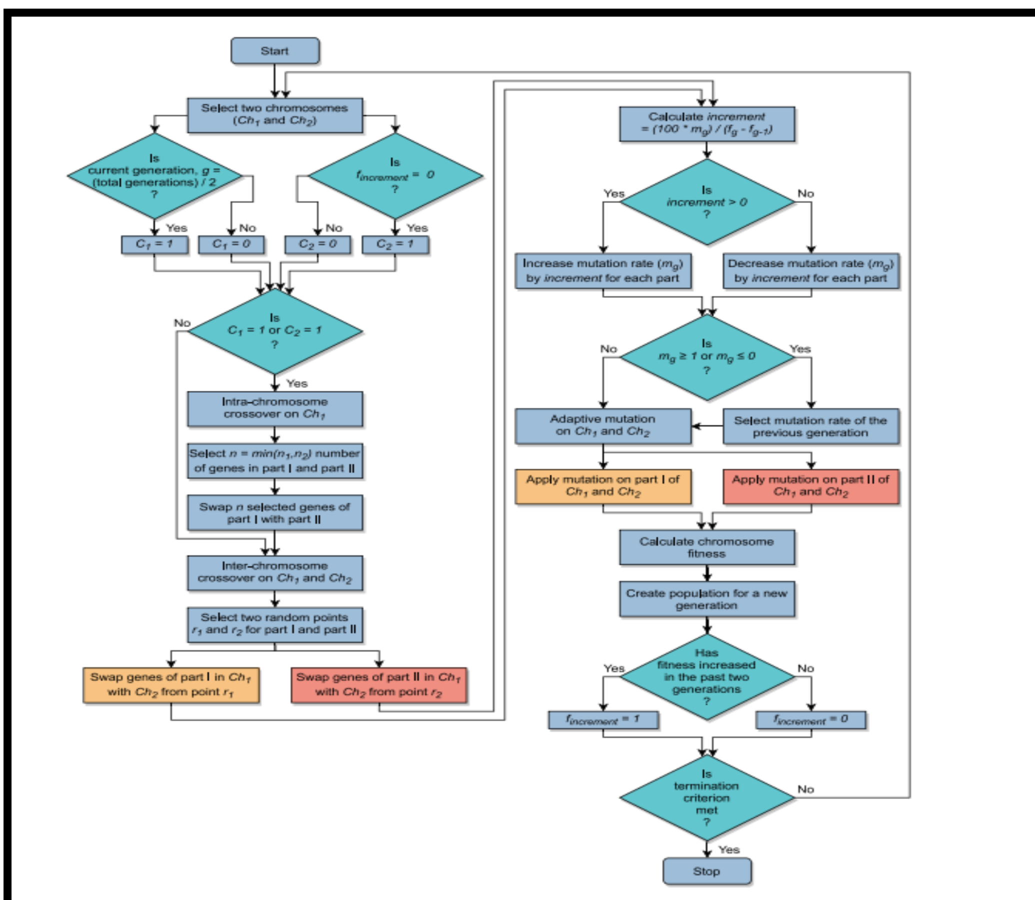
# Experiments

## Hyperparameters for Simple GA and GA ICAN:

- **Population size=10**
- **Chromosome length=17**
- **Generations=10**
- **Tournament size=2**
- **Crossover rate=1.0**
- **Inherited rate=0.5**
- **Mutation rate=0.05**

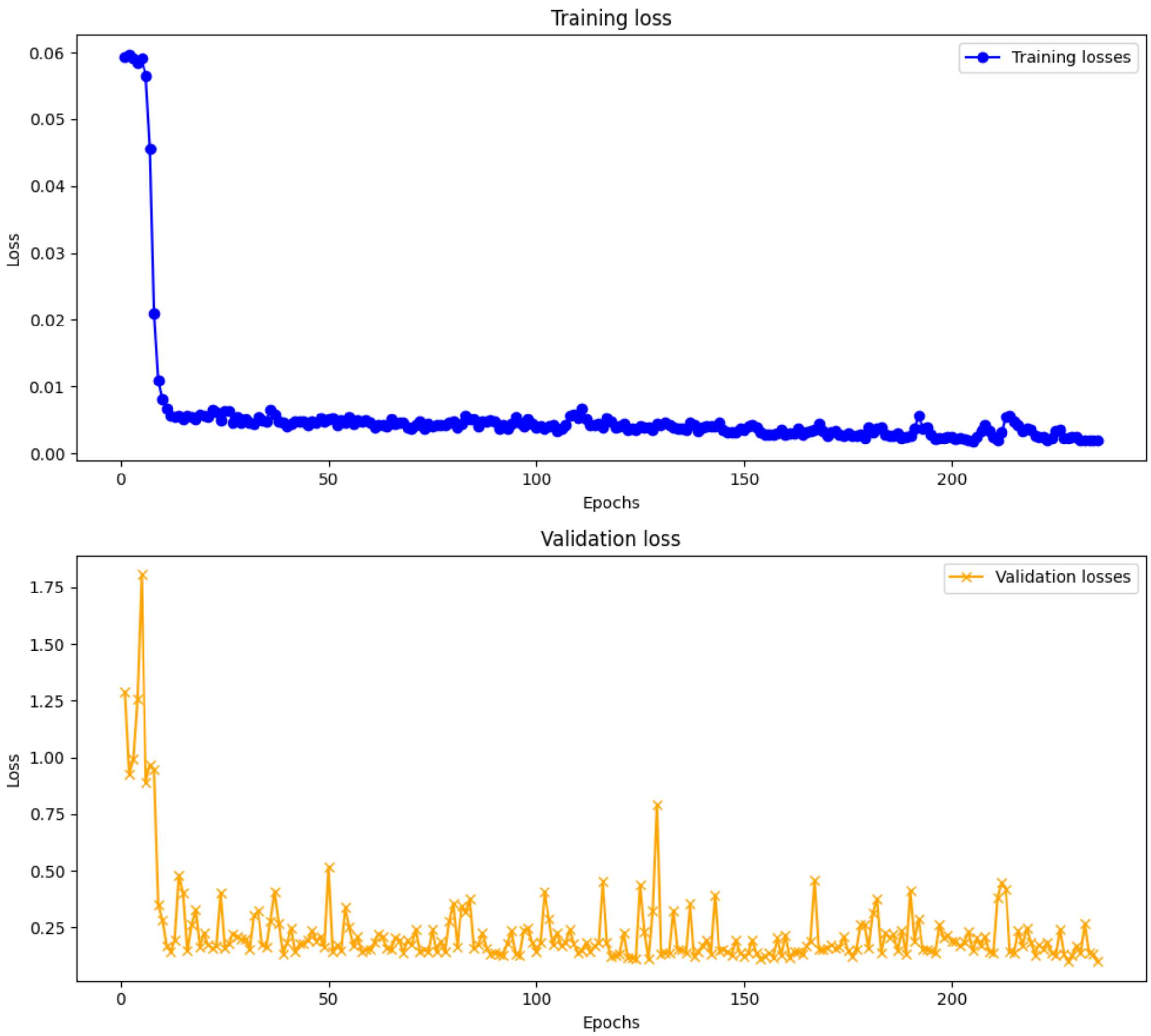
*(For a fair comparison to see which is the best for this specific problem I used the same hyperparameters in both GA algorithms models.)*

## ICAN GA flow chart



# Training and validation results

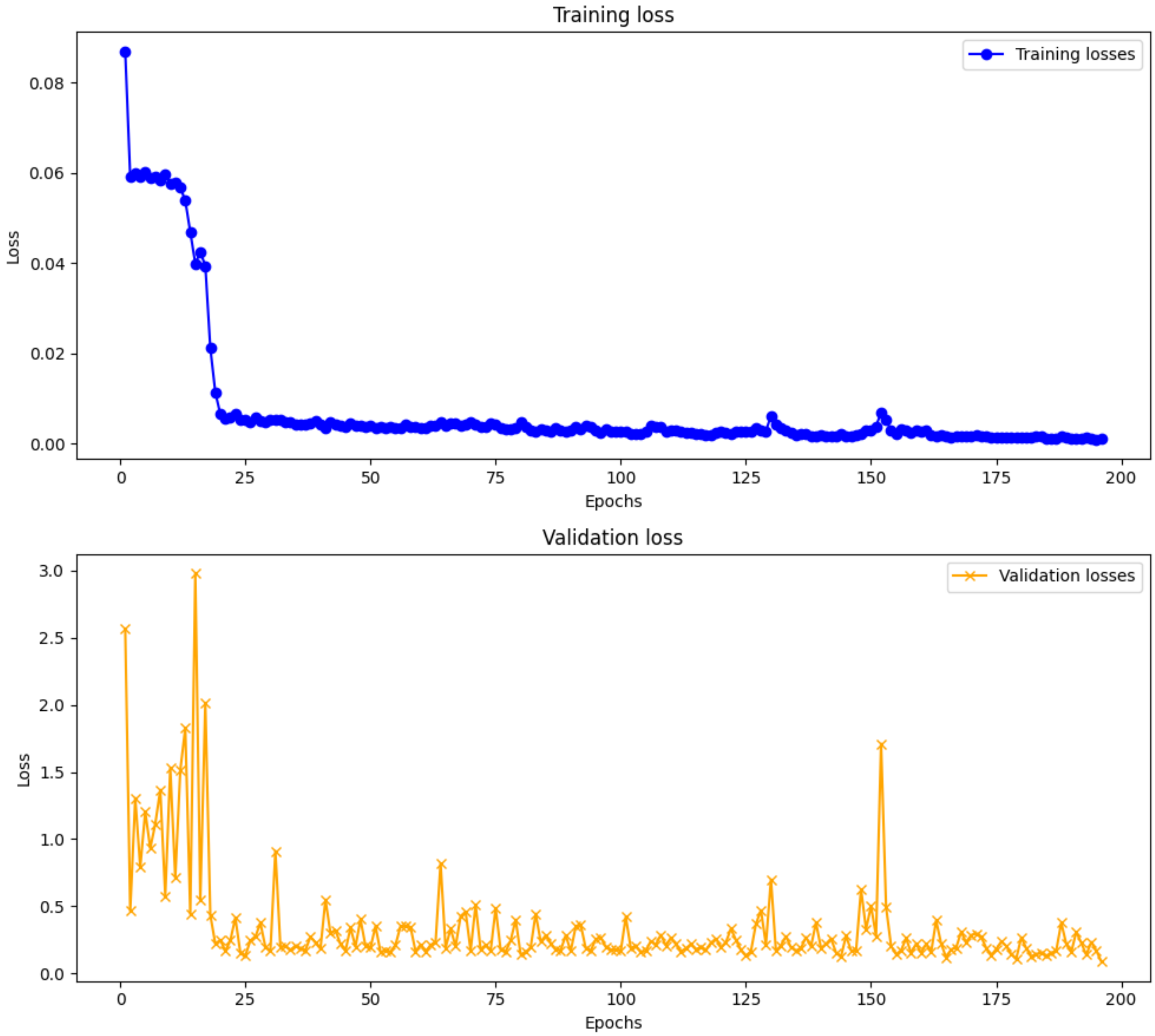
## Simple GA




GA - hidden_size=4, staked_layers=7, learning_rate=0.01, batch_size=8, epochs=500, split=0.9, loss=0.10062975925393403

The purpose of using **GA** according to the paper is to find the best hyperparameters for a **LSTM** model in order to **minimize the error** during the train and validation phase, as you can see in the image , a **simple GA** was able to find the best hyperparameters ***(You can see the best hyperparameters founded by simple GA in the image)*** minimize the error up to **0.10063** approximately.

# GA ICAN

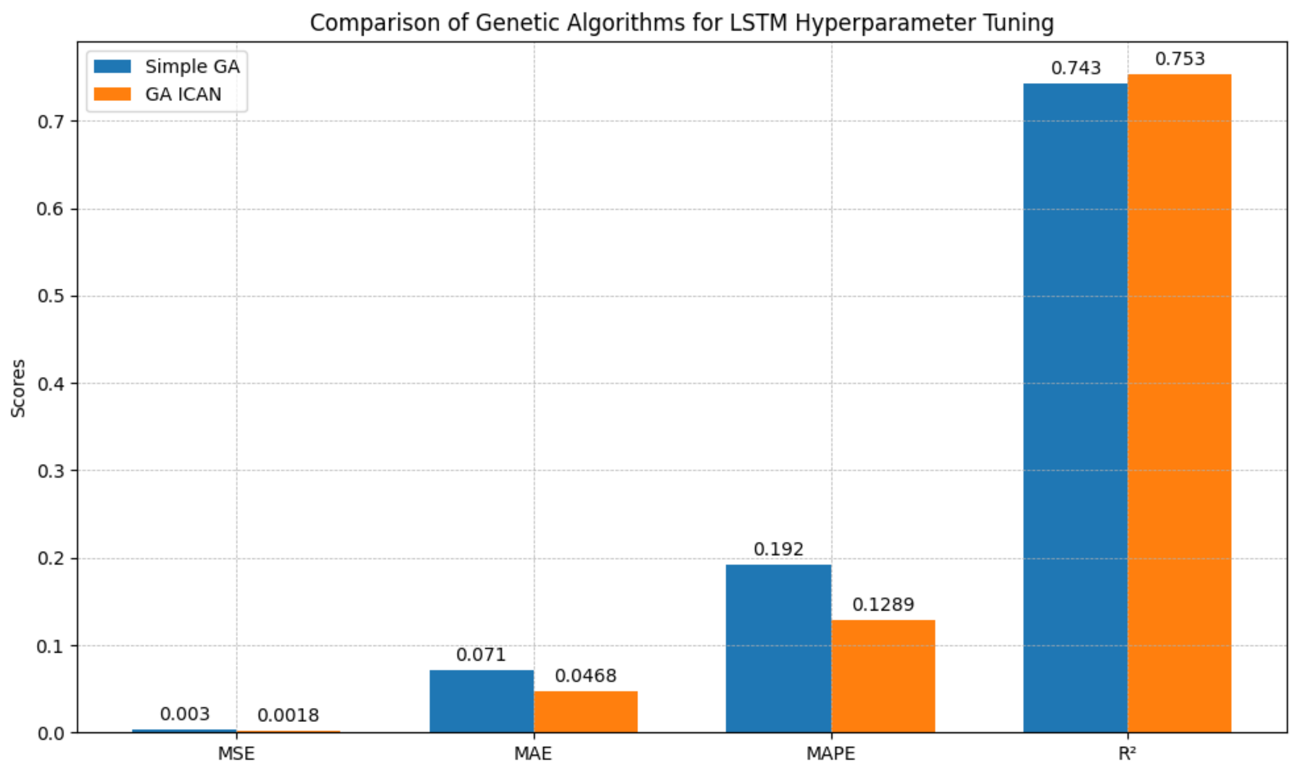



ICAN - hidden_size=6, staked_layers=8, learning_rate=0.01, batch_size=16, epochs=300, split=0.9, loss=0.0886113295564428

It can be noticed that **GA ICAN** surpassed the results of the **Simple GA,** It could find hyperparameters for **LSTM** model that got a **lower error** during training and validation phase up to **0.088611** approximately.

# Metrics optimization results

# Simple GA vs. GA ICAN

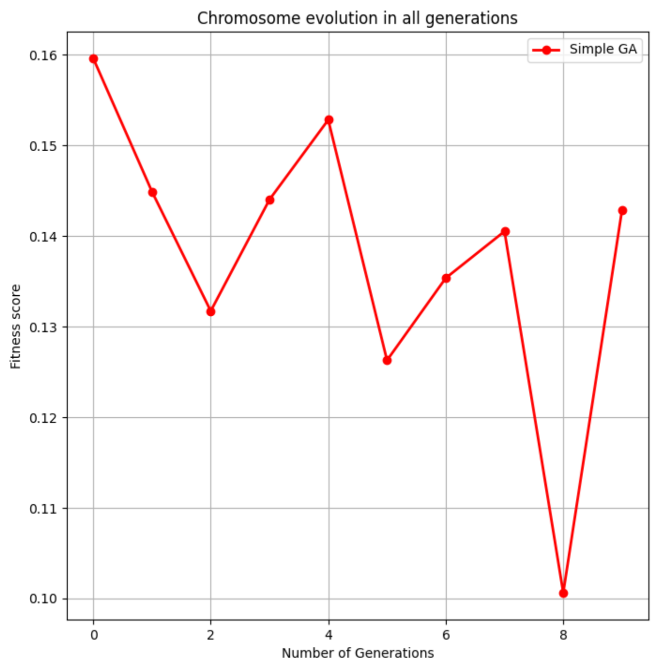


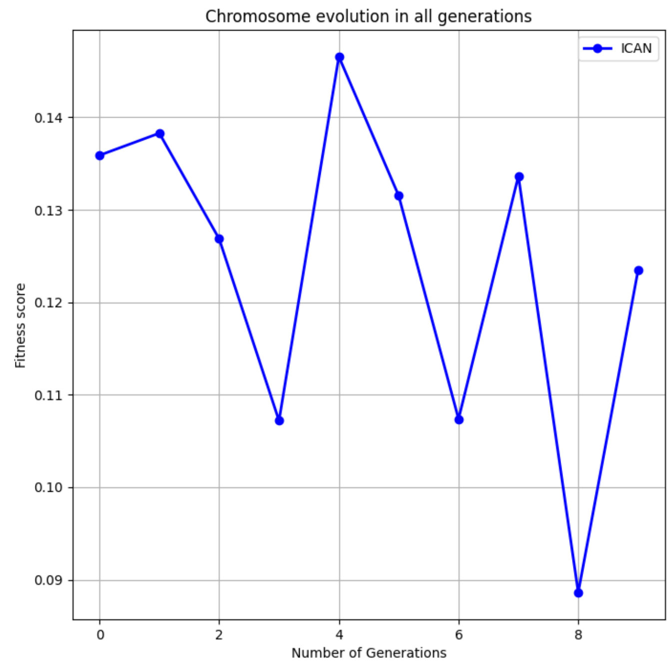

Comparison of Genetic Algorithms for LSTM Hyperparameter Tuning

| Metric | Simple GA | GA ICAN |
|---|---|---|
| MSE | 0.003019 | 0.001772 |
| MAE | 0.070968 | 0.046817 |
| MAPE | 0.191975 | 0.128946 |
| $R^2$ | 0.743038 | 0.75303 |

As can be observed, **GA ICAN** surpassed the **simple GA** in every metric **(MSE, MAE, MAPE)**. Since these are metrics for calculating the error of the model during the training and validation phase, **lower error is better**. For **$R^2$**, which stands for the **coefficient of determination, higher values are better.** $R^2$ measures the proportion of the variance in the dependent variable that is predictable from the independent variables. **A higher $R^2$ value indicates a better fit of the model to the data.**

# Performance of Simple GA vs GA ICAN




It can be observed that both algorithms presented a **high variability** in the solutions in each generation, which means that if maybe more generations can be added in order to find better results till both algorithms converge , however because these algorithms where use to train a **LSTM** model , **with only 10 Generations Simple GA trained the model more than 100 hundred thousand times** , since the **fitness function** was the loss **function MSE** ***(value than can only be obtained after the model is trained).***

## Execution time Simple GA vs GA ICAN


As can be observed even in the execution time **GA ICAN** was the winner. While **Simple GA** took **47.27 hours** , **GA ICAN** took **33.90 hours**. Since the goal was find the best  the hyperparameters for **LSTM** model , **GA** algorithms train the **LSTM** model every time the fitness is going to be evaluated in the GA operators , so practically with only **10 Generations Simple GA trained the model more than 100 hundred thousand  times** , since the **fitness function** was the loss **function MSE**, that's why in this kind of cases the comparison of both algorithms by multiple runs could take even weeks which is not efficient in time and resources.

## Conclusions

1. These experiments, consistent with the findings in the original paper, demonstrated that the GA ICAN algorithm is an effective method for identifying optimal hyperparameters for deep learning models. When compared to a Simple GA, GA ICAN consistently outperformed the Simple GA in every task, highlighting its superiority and robustness as a hyperparameter optimization technique.
2. Genetic Algorithms (GAs) are a valuable tool for optimizing deep learning models. When there is uncertainty about how to improve a model's performance, GAs can be considered a viable approach to achieve this goal.

## Discussion

1. After implementing this method myself, I have one observation: although GA ICAN is an effective method for optimizing the hyperparameters of a deep learning model, I believe its performance could be enhanced by using a type of GA called Messy GA. Messy GA first identifies relationships within the classes, which could potentially improve the algorithm's execution time. However, due to the limited time available for this final project, I was unable to implement this idea. I plan to explore this approach in future experiments.