Proc. of the International Conference on Electrical, Computer and Energy Technologies (ICECET)
9-10 December 2021, Cape Town-South Africa

Using Feature Engineering Techniques to Improve Multivariate Model Forecasting of the South African Unemployment Rate

Thabo Makola
School of Computer Science
And
Applied Mathematics
University of the Witwatersrand
Johannesburg, South Africa
736277@students.wits.ac.za

Rudzani Mulaudzi
School of Computer Science
And
Applied Mathematics
University of the Witwatersrand
Johannesburg, South Africa
rudzani.mulaudzi2@students.wits.ac.za

Ritesh Ajoodha
School of Computer Science
And
Applied Mathematics
University of the Witwatersrand
Johannesburg, South Africa
Ritesh.Ajoodha@wits.ac.za

***Abstract*—The unemployment rate in South Africa has been increasing yearly. Between January and March 2021, the unemployment rate has risen to 32.6%. This means that there are more than 7.8 million South Africans who are officially unemployed [1]. This paper explores feature engineering techniques that can be applied to improve the forecasting ability of multivariate machine learning forecasting models. Research has shown the ability of modern machine learning methods to forecast the unemployment rate in South Africa accurately. Few kinds of research have been conducted on improving these models. Feature engineering is one of the many ways that can be applied to improve the performance of these models. In this paper, we are looking at several feature engineering techniques such as combining, performing statistical calculations, and transforming features. Using SVR as the base model, the feature engineering techniques managed to improve the R^2 by over 80%. This paper presents feature engineering as a viable solution to improving the forecasting ability of multivariate forecasting machine learning models.**

***Index Terms*—Feature engineering, Forecasting, Machine learning, Unemployment rate**

I. Introduction

Feature Engineering is a machine learning process for extracting useful features out of raw data using domain knowledge. It aims to find significant features, through processes such as combining relatively insignificant features to contribute to the model. Consider predicting the unemployment rate in South Africa using time series data. This problem requires domain knowledge on what are the best features in predicting the unemployment rate and what features will most likely have no effect. An idea would be to select a batch of features randomly and try to model the problem using only those features. One would then try repeatedly taking a random sample of a set number of features until they find the model that best fits the data with low bias and high variance. This process of repeatedly taking samples is part of a class of methods under feature selection. These methods are used to select the best features that will improve the performance of the model in question. Consider another forecasting task: synthesizing new features to improve the performance of the model. You could take a small batch of features and see how they influence the model. Using the small sample, you can calculate statistics that tell how each feature behaves relative to the other. From this, you can notice that some pairs of features can be combined using certain mathematical tools to create a new feature. This feature might be more predictive than individual features. This can be true for a combination of n features where n represents the number of features in the sample. This process can be viewed as feature engineering. It is broader than feature selection and it is the main focus of this paper.

Previous work in improving the performance of multivariate machine learning forecasting models focused on feature selection techniques, which often do not cover the entire space of feature engineering, limiting the scope of the problem. Using feature engineering to improve the performance of the models, and keeping with the theme of forecasting the South African unemployment rate, each feature has the potential to be engineered to have a meaningful impact on the final model. Every feature in the data set is seen as useful rather than selecting only a subset and leaving other features unexplored. Feature engineering can generally be classified into eight categories. Binning, Log Transforms, One-Hot encoding, Scaling, Feature splitting, Grouping Operations, Handling outliers and Date Extraction.

We will briefly review the foundational concepts and techniques of feature engineering, discuss the type of data that is typically used in many forecasting problems, look into the type of models that can be used alongside feature engineering and what are the best measures of performance for forecasting problems in Section II.

978-1-6654-4231-2/21/$31.00

II. Related Work

South Africa's unemployment rate shows no sign of slowing down, in the fourth quarter of 2020 the rate was 32.5% the highest ever recorded [1]. This raises concerns regarding the well-being of the country. The South African market is still not creating enough jobs as more people enter the labour market. Accurately predicting the unemployment rate can help policymakers deal with the unemployment crisis. In forecasting unemployment, the most accurate models need to be utilized, but before this can be done, the best features for forecasting must be selected.

Most research on this topic has focused on improving the accuracy of forecasting models. Models are compared against each other and ranked according to their values of R^2, which is the most popular performance measure. Other performance measures utilised include Mean Absolute Scaled Error (MASE) and Mean Absolute Percentage Error (MAPE). A lot of work has not been done in optimizing features for unemployment forecasting. A small percentage of work has focused on feature selection techniques to improve forecasting. Much more research needs to be conducted on feature engineering techniques to assist the best performing models to improve their accuracy.

Most models cannot synthesis features that are based on ratio differences that well. Engineering such features will help the model perform better and this might increase the accuracy of the model. In this paper, we are looking at feature engineering techniques to improve the forecasting of multivariate models. The following section motivates and defends the proposed question.

A. Data

Data is one of the most important components of research. The most common datasets used for time series forecasting is usually around 100–160 features and over 700 observations [2]. The data is extracted from different sources, since our research question is addressing the issue of unemployment, the data is usually sourced from government sites. For forecasting of South African unemployment, the data is typically sourced from the Bureau of Economic Research, South African Reserve bank, Quantec, and Statistics South Africa. Sources for other research not predicting the South African unemployment rate include the Federal Reserve Bank of St Louis, Eurostat database etc. Due to the limited amount of research in South African unemployment rate forecasting, the data used has to be curated from the multiply sources mentioned above into one table. These sources house the data in different organizational structures, time frequencies, and number formats. For most of the literature excluding literature specific to South Africa, the data is usually model ready in a clean tabular structure.

The traditional models used to forecast the unemployment rate usually need the data to conform to certain restrictions for the models to perform well. Significant data wrangling is required to meet the model requirements. Approaches that take into account the specific characteristics of data are applied to offer more accurate forecasting. These approaches differ from dataset to dataset, but they can all be summarized as feature optimization techniques [3]. Long-memory and heteroskedasticity characteristics must be present in data if models such as FARIMA and FIRIMA/GARCH are to be used [4]. To show if the volatility of the previous period influences the volatility of the next period some research use the Ljung-Box test to check the heteroskedasticity of the data on squared residuals of a fitted FARIMA model. The R/S [5], the Higuchi [6] and the aggregate variance methods are the three methods used to detect long-memory using the Hurst exponent [7]. The method with the smallest values is used as a conservative approach. Primarily they are focusing on a Hurst exponent with a value in the range (0.5, 1) this suggests the existence of long memory with values closer to 1 suggesting stronger long memory. Other data characteristics that are part of improving the accuracy of prediction of models are the departure from the non-linearity and normality of the dependence structure of the data. For data that departures from normality models such as FARIMA and FARIMA/GARCH with non-normal errors are used. Non-linearity is accounted for by the use of models such as support vector machines, multivariate adaptive regression splines and neural networks if it exists [8].

Feature selection methods have also been shown to improve the accuracy of forecasting models. A couple of feature selection techniques have been applied in the literature about South African unemployment rate forecasting [9]. The first of the three feature engineering techniques is the *Filter methods*. This method uses statistical scores to rank features according to their weight in forecasting the target feature. The second is *Embedded methods*, which choose a subset of the features and measure their error rate marking the least impactful feature set as the one with the lowest error. The last ones are the *Wrapper methods*, These are less efficient compared to *Embedded methods* and can lead to combinatorial explosions since they evaluate every possible subset [10]. Most of the data contain missing values due to various reasons. The common reasons are the frequency of the reported data by different sources. Various missing data imputation techniques are used in literature [11] with the most common being *mean value imputation, constant imputation, forward imputation, k-nearest neighbour (kNN) imputation, multivariate imputation by chained equations, and last known value imputation*

B. Models

We are going to look into the models used in literature to forecast the unemployment rate or prediction of time series data. The models can be split into two camps, Statistical and Machine Learning models. Even though this paper is focused on improving Machine learning models. Engineering features benefits both camps. Traditional statistical models for modelling the unemployment rate and/or prediction of time series data are the **Naive** method which uses recently observed data to forecast the future. **ARIMA** model is the most commonly used statistical model [12], The model has been extended in other papers to model multivariate cases. There are two extended versions that are referred to by different names: **FARIMA** and **GARCH** [4]. Other methods are **Simple Exponential Smoothing** [13], **Holt-winters model**, and **Damped Exponential Smoothing** [14].

Statistical modelling has come a long way since its early days. In [15] machine learning methods are dominated by all the above mentioned statistical models. Most machine learning models are not as accurate as the naive model. This is mainly due to the number of empirical studies conducted to improve the accuracy of statistical these models. [16]. This is the major reason why statistical methods are still used for forecasting time series data.

There is a handful of machine learning models used to forecast unemployment rates and time-series data. This paper only focuses on the eight model families that are presented in the literature.

- **Multi-Layer Perceptron (MLP)** is a simple neural network composed of multiple layers of perceptrons.
- **Bayesian Neural Network (BNN)** similar to Multi-Layer Perceptron in that uses prior error distributions to estimate weights, i.e it uses Bayesian concepts to optimize the network parameters.
- **Radial Basis Functions(RBF)** The method is interpretable and faster to compute, it is very similar to MLP but it performs a linear combination of n basis functions that are radially symmetric around the centre instead of the sigmoid activation function.
- **Generalized Regression Neural Networks** Is an improvement in non-parametric regression neural networks in that each sample point represents an average to a radial basis neuron. The advantage of GRNN is that back-propagation is not needed since it is single-pass learning.
- **K-Nearest Neighbor regression.** KNN averages the observations in the same neighbourhood to approximate the association between the continuous outcome and independent variables. It uses cross-validation to select the size that reduces the mean square error when setting the size of the neighbourhood.
- **CART regression trees (CART)** are tree-like partitioning of the input space recursively. Based on the input which is the training sample divided into regions. Items provided are classified using a sequence of tests introduced and applied to decision nodes.
- **Support Vector Machine.** The idea behind SVR is to find an N-dimensional hyperplane that can correctly classify data.
- **Gaussian Processes (GP)** predict the future value of interest by combining input points using a measure of similarity and assuming an a priori distribution for the input variables provided during training. It serves as a non-parametric regression method.

There is a disagreement in the literature on whether machine learning models can hold their own and perform better than statistical methods. With some literature claiming that most machine learning methods models still have to go a long way to achieve an accuracy of statistical methods.

C. Feature Engineering

A couple of studies suggest feature optimization methods to improve accuracy. One such method is feature engineering is a process that involves extracting features from raw data. These are properties or characters that are shared by various units within a domain. The common feature engineering techniques are *Imputation*, Missing values are often encountered when preparing data for machine learning. They can be caused by human errors, network issues, or privacy concerns. Some machine learning platforms try to automatically drop the rows that contain missing values during the model training phase. This causes the model to perform poorly and has reduced performance. There are two types of imputation

1) *Categorical Imputation* If the missing values are in a column, then replacing them with the maximum occurred value is a good idea. But, if the values are distributed uniformly, then imputing a category like Other might be more sensible.
2) *Numerical Imputation* is a more advantageous option since it allows you to preserve the data size while also reducing the number of columns that contain missing values.

Binning, The main goal of binning is to improve the model's robustness and prevent over-fitting. However, this is not very cost-effective and it can also affect the performance of the model. Generally, if your data has low frequencies, it might be better to assign a general category to the labels.
Log Transform is a commonly used feature engineering technique to handle skewed data. It helps minimize the effects of outliers and increases the efficiency of the model. In most cases, the magnitude order of the data does not change after the transformation. Also, it decreases the dependence on the outliers.
One-Hot encoding One-hot encoding is an important technique in machine learning. It spreads the values of a column to multiple flags and assigns one or more zeros to those flags. This method allows you to group your data without losing any information.
Scaling Most of the time, the numerical features of a dataset do not have a defined range. Instead, they vary from each other. This issue can be solved by simply scaling the continuous features. There are two types of scaling

1) Standardisation takes into account standard deviation, z-score normalization (standardisation) scales the values while keeping in mind the differences between the features.
2) Normalization refers to a process that adjusts the values in a fixed range. It does not affect the distribution of the feature and is not required to handle the outliers.

Feature Split Splitting features can help improve the performance of machine learning algorithms by making them aware of the unused parts of a column. They can then be used to improve model performance by discovering new information.
Grouping Operations In most cases, the dataset is represented by a row, where every column shows a different feature of the

instance (Tidy). A dataset that has multiple rows of instances is not a tidy data example. Instead, it should be grouped by its instances and only one row is used for each instance.

Handling Outliers The best way to detect outliers is by showing the data visually. This method gives a chance to decide with high precision. Here applying statistical techniques can identify outliers much faster.

Extracting date Though date columns are usually used for model target validation, they are rarely utilized by algorithms for machine learning. Before you start working on a machine learning algorithm, make sure that the date columns are pre-treated to make them easier to understand. Doing so can help avoid generating an ordinal relationship between the values.

It is noted that Features engineered based on ratio differences were not well grouped by any of the models explored in [17]. Due to the ratio of difference, descent-based features can be useful for various models. For example, support vector machines and neural networks can benefit from these features. Also, count-based engineering features are very helpful for random forests and reinforcement machines.

D. Measures of Performance

Which models use what measure of accuracy changes a lot of things. It is important to choose the best performance measure as this can make or break your results. Different papers use different measures of accuracy to assess the performance of models. The most popular performance measure is The R^2. It evaluates the ability of the model to explain the variation in the data. In other words, it answers the question "How relevant is the model to the data?" [18]. We also have a common performance measure in time-series analysis, The Mean Absolute Percentage Error (MAPE). Since MAPE penalizes large positive errors more than negative ones [19], MASE is usually introduced as a band-aid to this problem [20]. Mean Absolute Scaled Error (MASE) among its other characteristics is data scale-independent. Its value is greater than one if it is less accurate than the average model fitting prediction of the seasonal Naive benchmark, and less than one if the forecast is more accurate.

E. Conclusion

Statistical forecasting has progressed a lot since its early days. This progress is due to the demand for empirical studies by both practitioners who want to utilise the most accurate models as well as the academic community. These studies helped establish two ideas in the statistical world. Firstly, models that best fitted sampled data did not necessarily fit real-world data that well. Secondly, the post sample forecasting of simple models is as good as the sophisticated ones. In [15], we find that the simple statistical methods are as accurate as the sophisticated machine learning ones. In [21] the opposite is true, machine learning models perform better but not by a wide margin, the MLP's MASE and R^2 are 2.387 and 0.696 while the ARIMA achieves similar results 2.391 and 0.696. The clear advantage that the statistical models have over the machine learning ones is that they are computational less intensive. Much has to be done to improve the forecasting ability of machine learning models. While providing new machine learning models that can forecast better than the currently available models can be a solution. This approach is time-consuming and very costly. At the same time, empirical proof needs to be provided that test these models against benchmarks such as the naive method. A lot of machine learning forecasting literature lack such proof. Another problem faced by the academic literature of machine learning models is the reliance on sophisticated machine learning models while simple models can do as good of the job.

One area that shows light in improving the accuracy of machine learning models is feature optimisation techniques. [9] applied feature selection on multivariate deep learning machine learning models and demonstrated the forecasting ability of these models. [22] applied feature engineering techniques and saw the R^2 improve by over 100% on average, the MASE also decreased by approximately 1% proving that these techniques work. Machine Learning Forecasting academic literature lacks feature engineering implementation which can be used to improve the forecasting ability of most machine learning models.

III. Results and Discussions

A. Data Preparation

The Data used in this paper was sourced from the South African Reserve Bank. It consists of 794 rows and 47 features. The features can be categorized as monthly, quarterly and annually. Each feature is a time series from 1960 January to December 2019. The data had missing values due to mixed frequencies. The issue was addressed by deleting quarterly and annual data. To further address missing values in the monthly data, 7 imputation techniques were applied to the data. The techniques are as follows.

Filling missing values with 0's, this method was preferred over removing rows with missing values to preserve the size of the dataset. This data set gave an R^2 score of 0.274 which was the second-highest score. It was also used as the benchmark.

Forward fill this paper [21] proved the method to be more effective than other methods. This paper also gave the highest R^2 score of 0.496. The data set resulting from this method was used as a base for other engineering techniques.

Other methods included. *Univariate feature imputation* which imputes values in the i-th feature dimension using only non-missing values in that feature dimension. Under this class, we explored *mean, median and most frequent* imputation techniques.

Multivariate feature imputation which uses the entire set of available feature dimensions to estimate the missing values.

Nearest neighbours imputation for filling in missing values using the k-Nearest Neighbors approach. Each method was evaluated using a 5-fold cross-validation technique. Table I shows the R-squared score for each k-th fold.

Figure show the highest R-squared score for each imputation technique. For each k-th fold the testing set was restricted to

TABLE I

R-SQUARED SCORES FOR EACH IMPUTATION METHOD PER K FOLD

k-th	benchmark	mean	ffill	median	KNN	multi	most frequent
0	0.209985	-0.214164	-0.532729	-0.411825	-0.214164	-0.214164	-0.524148
1	0.202815	0.155969	-0.24731	-0.00322	0.155969	0.155969	-0.582374
2	0.078732	0.004216	0.09134	-0.164357	0.004216	0.004216	-0.802993
3	0.41782	0.037753	-0.612516	-0.066482	0.037753	0.037753	-0.985309
4	0.274759	-0.056978	0.496297	-0.101263	-0.056978	-0.056978	-1.307616

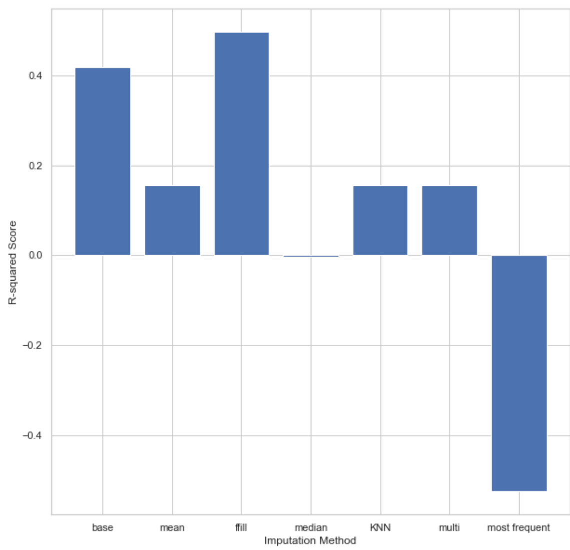

Fig. 1. Highest R-squared Scores for different imputation methods

24 while the training set kept changing due to out data being time series data and can not be split at random.

B. experiments

SVR was used as the model for testing various techniques. The choice was motivated by the fact that it had the highest R-squared score as compared to other models when feature selection was applied. The process of engineering features can be summarised into four categories.

1) **Lags** This process explores the impact of adding lags to the data. The focus is on how many lags are best suited to improve the model forecasting ability. The paper [21] outlined that two lags showed performance improvement as opposed to the other number of lags. For this experiment, two lags were added to the data.
2) **Statistical Features** These features were sourced using statistical measures. For each feature in our base dataset. The following features were engineered.
   - mean
   - standard deviation
   - average absolute deviation
   - minimum value
   - maximum value
   - median
   - median absolute deviation
   - interquartile range
   - negative values count
   - positive values count
   - skewness
   - kurtosis
   - energy

   Most of these are self-explanatory. A rolling window of size 7 was used in generating these features. Since the rolling window generates *NAN* values. These were filled with zeros. In total $47 \times 14 = 611$ new features.
3) **Fast-Fourier transform (FFT)** The Fourier transform is a function that takes time-domain signals into the frequency domain. These features were created by taking the Fourier transform of the base dataset and applying statistical transformations to obtain a total of 611 features.
4) **tsFresh features** Before these features were extracted a correlation between the original 47 features and the target variable revealed the most highly correlated feature. This feature is *Domestic output: All groups* Using the python package tsFresh, 787 features were extracted from *Domestic output: All groups* feature. Extracting features only on this feature and not on the other 46 was motivated by the fact that the correlation of the other 46 was less than $|0.5|$. It was wiser to pursue the features with a correlation of above 0.9.

The total number of features that were engineering ended up at 3237. Feature selection was applied to reduce the dimensionality of the problem. Model features selection was applied with the model of choice being *linear regression*. This reduced the number of new features to a value of 359.

C. Analysis and Results

The first feature engineering technique that was applied in this paper was imputation. five imputation methods namely Forward fill, mean imputation, median imputation, most frequent imputation, multivariate imputation and KNN imputation were applied to the original data. The data generated from each of the different imputation methods were divided into five cross-validation sets. SVR was run on each dataset and the R^2 score of the model for each set was recorded. Figure 1 shows the R^2 scores of each imputation method. Forward fill imputation gave the best R^2 score of 0.496. The lowest forward fill score recorded is -0.61, and the lowest score overall was -0.67 and comes from Multivariate imputation. The highest score obtained by Multivariate imputation is -0.015. Table I shows the score for each cross-validation set per imputation method.

Following imputation techniques, experiments were run on the 359 features that were extracted across the different categories. Each feature was added to the base data set one at a time and an SVR model was run on it after the addition of the feature. Then the feature will be removed and a new feature would be added in its place. This was done to measure the impact of each feature on the dataset. Across all four categories of selected features, the tsFresh category had 272 features out of 359 features. The average R^2 score for this category is 0.547083. The feature that improved the base data set the best also came from this category with an R^2 score of 0.580946. The worst performing category is Fourier Transform Features. No features from this category were selected. This could be the result that these features have little to no correlation with our target variable. Table II shows the statistics of each category. The graph shows the top 3 features relative to the rest of the features which are labelled as other. The lowest R-squared score recorded is 0.477384, this is lower than our base data set score with no added features.

TABLE II
STATISTICS OF EACH CATEGORY

category	count	mean	std	min	25%	50%	75%	max
lag features	1.0	0.547083	NaN	0.547083	0.547083	0.547083	0.547083	0.547083
stats features	86.0	0.506650	0.020977	0.456064	0.492366	0.505571	0.518978	0.571257
tsFresh features	272.0	0.477384	0.046909	0.270270	0.465726	0.492038	0.498205	0.580946

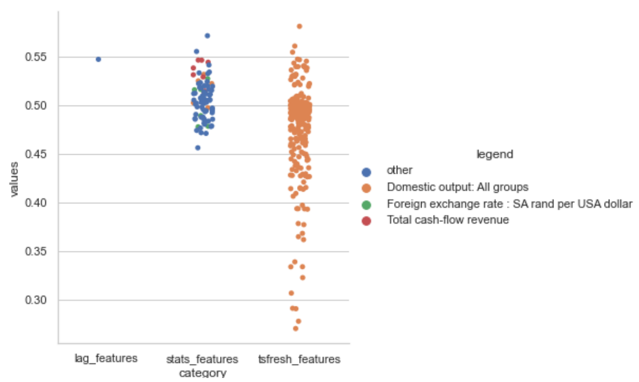

Fig. 2. Distribution of R^2 scores for each category

IV. Conclusion

This paper has shown that feature engineering techniques are viable tools to consider when forecasting time series data. From the benchmark, the dataset achieved a score of 0.274. Feature engineering techniques such as imputation improved the score by 81%. With the addition of just one engineered feature, the score improved by a further 1.8%. This research only used the addition of one feature to improve the forecasting ability of the SVR model. Future research could explore other models and the addition of a combination of other features.

Acknowledgment

This work is based on the research supported in part by the National Research Foundation of South Africa (Grant numbers: 121835).

References

[1] statssa, "Quarterly labour force survey," http://www.statssa.gov.za/publications/P0211/P02112ndQuarter2021.pdf.

[2] A. Ajiboye, R. Abdullah-Arshah, and Q. Hongwu, "Evaluating the effect of dataset size on predictive model using supervised learning technique," 2015.

[3] J. Lu, K. Plataniotis, and A. Venetsanopoulos, "Face recognition using feature optimization and/spl nu/-support vector learning," in *Neural Networks for Signal Processing XI: Proceedings of the 2001 IEEE Signal Processing Society Workshop (IEEE Cat. No. 01TH8584)*. IEEE, 2001, pp. 373–382.

[4] C. Katris, "Prediction of unemployment rates with time series and machine learning techniques," *Computational Economics*, vol. 55, no. 2, pp. 673–706, 2020.

[5] B. Mandelbrot, "Statistical methodology for nonperiodic cycles: from the covariance to r/s analysis," in *Annals of Economic and Social Measurement, Volume 1, Number 3*. NBER, 1972, pp. 259–290.

[6] J. Beran, R. Sherman, M. S. Taqqu, and W. Willinger, "Long-range dependence in variable-bit-rate video traffic," *IEEE Transactions on communications*, vol. 43, no. 2/3/4, pp. 1566–1579, 1995.

[7] H. E. Hurst, "Long-term storage capacity of reservoirs," *Transactions of the American society of civil engineers*, vol. 116, no. 1, pp. 770–799, 1951.

[8] B. Wightman, I. Ha, and G. Ruvkun, "Posttranscriptional regulation of the heterochronic gene lin-14 by lin-4 mediates temporal pattern formation in c. elegans," *Cell*, vol. 75, no. 5, pp. 855–862, 1993.

[9] R. Mulaudzi and R. Ajoodha, "Application of deep learning to forecast the south african unemployment rate: A multivariate approach," in *The 7th Asia-Pacific Conference on Computer Science and Data Engineering*, 2020.

[10] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16–28, 2014.

[11] D. Bertsimas, C. Pawlowski, and Y. D. Zhuo, "From predictive methods to missing data imputation: an optimization approach," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 7133–7171, 2017.

[12] R. J. Hyndman, Y. Khandakar *et al.*, "Automatic time series forecasting: the forecast package for r," *Journal of statistical software*, vol. 27, no. 3, pp. 1–22, 2008.

[13] E. S. Gardner Jr, "Exponential smoothing: The state of the art," *Journal of forecasting*, vol. 4, no. 1, pp. 1–28, 1985.

[14] ——, "Exponential smoothing: The state of the art—part ii," *International journal of forecasting*, vol. 22, no. 4, pp. 637–666, 2006.

[15] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "Statistical and machine learning forecasting methods: Concerns and ways forward," *PloS one*, vol. 13, no. 3, p. e0194889, 2018.

[16] R. G. Brown, *Statistical forecasting for inventory control*. McGraw/Hill, 1959.

[17] J. Heaton, "An empirical analysis of feature engineering for predictive modeling," in *SoutheastCon 2016*. IEEE, 2016, pp. 1–6.

[18] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*. OTexts, 2018.

[19] P. Goodwin and R. Lawton, "On the asymmetry of the symmetric mape," *International journal of forecasting*, vol. 15, no. 4, pp. 405–408, 1999.

[20] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International journal of forecasting*, vol. 22, no. 4, pp. 679–688, 2006.

[21] R. Mulaudzi and R. Ajoodha, "An exploration of machine learning models to forecast the unemployment rate of south africa: a univariate approach," in *2020 2nd International Multidisciplinary Information Technology and Engineering Conference (IMITEC)*. IEEE, 2020, pp. 1–7.

[22] ——, "Improving the performance of multivariate forecasting models through feature engineering: a south african unemployment rate forecasting case study," 2021.