RAFE-Net: A Residual Adaptive Residual Ensemble Feedback Network to Improve Accuracy of Prediction

[1]S. Gladson Oliver, [2]T. Suguna, [3]C. Aswini, [4]R. Malavika
[1]Assistant Professor, [2]Assistant Professor, [3]Assistant Professor, [4]Assistant Professor
Department of Information Technology,
Government College of Technology, Coimbatore, India.

***Abstract*—Prediction systems are the most important part of making decisions based on data. They affect areas like finance, healthcare, industrial automation, and climate science. Even with improvements in deep learning and machine learning, current predictive models still have three big problems: they make more mistakes with each prediction, they are sensitive to changes in the data distributions they are based on, and they can't always capture how different features interact with each other. This paper presents RAFE-Net (Residual Adaptive Feedback Ensemble Network), an innovative algorithm specifically formulated to tackle these constraints. RAFE-Net uses the best parts of ensemble learning, a residual feedback module (RFM) that learns from mistakes all the time, and a distribution shift detector (DSD) that changes predictions when data distributions change a lot. The framework not only makes things more accurate, but it also makes them more robust and easier to understand. This makes it good for high-stakes situations like fraud detection and medical diagnosis. Experimental assessments utilising benchmark datasets—such as the M4 time series dataset, the IEEE-CIS fraud detection dataset, and various datasets from the UCI Machine Learning Repository—illustrate that RAFE-Net attains a predictive accuracy improvement of up to 7.2% and a reduction in false positives by 12.5% relative to leading-edge baselines. These findings underscore the promise of feedback-driven ensemble frameworks as the forthcoming generation of predictive modelling systems.**

***Index Terms*—Prediction accuracy, residual learning, ensemble models, adaptive feedback, time series analysis, anomaly detection.**

I. INTRODUCTION

Prediction is now a key part of modern data-driven decision-making systems that are used in many fields and industries. In finance, predictive algorithms help find fake transactions, figure out credit risks, and predict stock market trends. In healthcare, prediction models help with early disease diagnosis, personalised medicine, and the best use of resources. Climate scientists use predictive analytics to figure out what the weather will be like, when natural disasters will happen, and what climate change will do in the long term. Predictive maintenance cuts down on downtime and makes things run more smoothly in factories and other industrial settings. In all of these areas, predictive systems' accuracy and reliability are not only technical achievements, but they also have a big effect on society and the economy [1], [2].

Despite decades of progress in predictive modelling, there are still some problems with current methods. Standard statistical models like autoregressive integrated moving average (ARIMA) [3] and linear regression work well with structured and stationary datasets. But they assume linearity and don't work as well when the data has non-linear patterns or interactions between many features. Machine learning techniques, such as Random Forests [4], Gradient Boosted Machines [12], and Support Vector Machines, enhanced predictive accuracy by assimilating intricate decision boundaries. Still, these algorithms usually treat each prediction as separate and don't have ways to learn from mistakes made in earlier predictions.

The rise of deep learning led to architectures like Long Short-Term Memory networks (LSTMs) [5] and Transformer-based models [6] that can capture long-range feature interactions and temporal dependencies.

These models are very accurate, but they still need a lot of data to work well and can slow down when the distribution of real-world datasets changes [8], [9]. For instance, a fraud detection model that was trained on data from last year may not be able to find new fraud schemes that take advantage of new weaknesses. Predictive healthcare models may also fail when the demographics of patients or the distribution of diseases change over time.

These problems show three long-lasting gaps in predictive modelling. First, existing algorithms don't pay much attention to how errors build up over time. A system that makes wrong predictions early in a forecasting window often keeps making wrong predictions because it doesn't learn from its mistakes. Second, changes in data distribution make models less useful in general. Retraining strategies are a big part of how things are done now, but they are expensive and don't always happen on time. Third, ensemble models bring together several learners to improve performance, but they don't often adapt to capture complex higher-order feature interactions [10]. Addressing these gaps is critical to advancing predictive analytics.

In response, this paper proposes RAFE-Net (Residual Adaptive Feedback Ensemble Network), a novel prediction framework that integrates ensemble learning with residual error correction and adaptive recalibration under distribution shifts. This work makes three unique contributions: (1) it introduces a Residual Feedback Module (RFM) that learns from past prediction errors using attention-based methods; (2) it adds an Adaptive Correction Layer (ACL) that uses these residuals to make final predictions; and (3) it adds a Distribution Shift Detector (DSD) that keeps an eye on statistical divergence in feature distributions and adjusts models in changing environments. These innovations tackle the problems of error accumulation, shifts in distribution, and complicated interactions between features.

The rest of this paper is set up like this: Section 2 looks at other work that has been done in predictive modelling, including statistical, machine learning, and deep learning methods, as well as residual learning and adaptation to distribution shifts. Section 3 goes into detail about the RAFE-Net framework and its parts. In Section 4, we talk about the experimental setup, which includes the datasets, baselines, and evaluation metrics. Section 5 shows the results and analysis, along with comparisons to the best baselines. Section 6 wraps up with talks about contributions, limitations, and ideas for future work.

II. Related Work

The quest for precise predictive models has developed over several decades, integrating insights from statistics, machine learning, and deep learning. This section looks at the state of the art and points out where current methods work well, where they don't, and how these problems led to the design of RAFE-Net.

Statistical Prediction Models

Statistical models have been the basis for predictive analytics. Autoregressive Integrated Moving Average (ARIMA) [3] and its variants are commonly employed in economic forecasting, demand prediction, and meteorology. Their strength comes from being simple and easy to understand. But ARIMA assumes that things are linear and stationary, which means it doesn't work with datasets or systems that are not stationary or have non-linear dependencies. Seasonal ARIMA (SARIMA) tries to account for seasonality, but it still has trouble with sudden structural changes [11]. Regression models like linear regression and logistic regression are also used a lot, but they have trouble with multicollinearity and non-linear interactions between features [1].

Machine Learning Approaches nits

Machine learning gave us more freedom than linear models. Decision Trees, Random Forests [4], and Gradient Boosting Machines (GBMs) [12] are great at finding relationships and interactions that aren't straight lines. Random Forests are strong against overfitting, but they can be hard to understand and may not work as well with very unbalanced data [10]. Gradient boosting techniques, including XGBoost [13], LightGBM, and CatBoost, prevail in structured data prediction contests. These algorithms are strong, but they assume that samples are independent and don't have clear ways to add sequential dependencies or fix errors. When data distributions change, they also need to be retrained, which can take a lot of time and money.

Deep Learning Models

Deep learning has changed how predictive modelling works. Long Short-Term Memory (LSTM) networks and Recurrent Neural Networks (RNNs) [5] deal with time dependencies, which is why they are widely used in speech recognition, financial forecasting, and medical sequence analysis. But LSTMs can be very expensive to run and need a lot of training data. Transformer architectures [6] brought in self-attention mechanisms, which made it possible to do parallel computation and model long-range dependencies well. Transformers have reached the highest level of performance in natural language processing and are being used more and more to predict time series [16]. However, deep learning models frequently require substantial data, are susceptible to overfitting, and are at risk of dataset shift [8], [9]. Also, their black-box nature makes them hard to understand, which is very important in fields like finance and healthcare.

Residual Learning

Residual learning came about with the ResNet architecture [18], which showed that residual connections improve gradient flow and model stability in computer vision tasks. Residual methods have been modified for sequential and tabular data; however, the majority of implementations prioritise training stabilisation over error feedback aimed at enhancing predictive accuracy [19]. The systematic incorporation of residual mechanisms for the identification and rectification of prediction errors is still insufficiently examined, presenting an opportunity for innovative architectures such as RAFE-Net.

Distribution Shift Detection

In practical predictive tasks, data distributions often exhibit dynamic characteristics. When the training and deployment data are very different from each other, this is called a dataset shift, and it can make performance worse [8]. To find drift, people use methods like watching the Kullback-Leibler (KL) divergence or the Population Stability Index (PSI) [20]. Recent research has suggested adversarial and Bayesian methodologies to enhance drift detection [17]. Despite these advances, most predictive systems rely on periodic retraining rather than continuous adaptation, resulting in delayed response to emerging changes.

Research Gap

From the literature, three major gaps emerge: (1) predictive systems rarely use residual error feedback to refine predictions, (2) robust adaptation to distribution shifts is not standard practice in most models, and (3) ensemble methods lack dynamic mechanisms to model higher-order feature interactions. The main goal of RAFE-Net is to fill in these gaps. RAFE-Net connects the gap between theoretical and practical robustness in predictive analytics by using ensemble learning, residual feedback, and shift detection.

III. RAFE-NET: PROPOSED METHOD

The suggested Residual Adaptive Feedback Ensemble Network (RAFE-Net) aims to solve three main problems in predictive modelling: too many errors, changes in data distribution, and not enough modelling of how features interact. The framework combines ensemble learning with a way to find and fix residual errors and a way to find changes in the distribution. This section talks about the structure, main parts, and math behind RAFE-Net.

Architectural Overview

RAFE-Net is made up of four main parts: (1) Base Ensemble Layer (BEL), (2) Residual Feedback Module (RFM), (3) Adaptive Correction Layer (ACL), and (4) Distribution Shift Detector (DSD). The BEL collects predictions from different base learners, like Gradient Boosted Trees, LSTMs, and Transformers, to make sure that both structured and temporal data patterns are covered. The RFM keeps track of residuals, which are the differences between predictions and the real thing. It also uses an attention mechanism to learn how errors happen in a systematic way. The ACL uses these corrections to change predictions in real time. Finally, the DSD keeps an eye on statistical differences in input distributions and adjusts the model when big changes happen.

Figure 1 shows the high-level structure of RAFE-Net. The BEL processes input data to make initial guesses. To find residual errors, these predictions are compared to ground truth labels. The RFM uses these residuals from the past few time windows to make corrective signals. The ACL uses these signals to make current predictions more accurate. The DSD also uses KL divergence and the Population Stability Index to look for changes in the distribution of input features. The system automatically changes thresholds or retrains parts of the RFM when it finds a drift.

Base Ensemble Layer (BEL)

BEL is the base of RAFE-Net. It combines different predictors to get the best of statistical, machine learning, and deep learning models. For instance, Gradient Boosted Machines are great at structured tabular data, LSTMs are great at capturing sequential dependencies, and Transformers are great at modelling long-range relationships. You can set up the ensemble as a weighted average or by stacking, with a meta-learner figuring out the best weighting.

Residual Feedback Module (RFM)

The RFM is what makes RAFE-Net different from regular ensembles. It explicitly models residuals, which are defined as $e_t = y_t - \hat{y}_t$, where y_t is the actual outcome and $\hat{y}_t$ is the predicted value at time t. The RFM doesn't throw away mistakes after checking them; instead, it sees them as useful signals. The RFM learns systematic error patterns by looking at sequences of residuals. For example, it learns that it often underpredicts during seasonal peaks or misclassifies minority classes. The RFM's attention mechanisms weigh residuals based on how relevant they are to the time period, which lets the system focus on recent errors while still taking longer-term patterns into account.

Adaptive Correction Layer (ACL)

The The ACL combines corrections made by the RFM into the ensemble's predictions. If δ_t is the corrective factor that the RFM gives at time t, then the corrected prediction is:

$$\hat{y}_t(new) = \hat{y}_t + \delta_t.$$

This system makes sure that predictions change based on past mistakes, which lowers the chance of making the same mistake again. Depending on the needs of the application, the ACL can be implemented with linear combination, gated fusion, or neural integration layers.

Distribution Shift Detector (DSD)

The DSD makes sure that the system is strong in environments with non-stationary data. It uses KL divergence to keep an eye on feature distributions in streaming or batch data:

$$D_{KL}(P \parallel Q) = \Sigma\, P(i) \log (P(i)/Q(i)),$$

where P is the training distribution and Q is the current distribution. If D_{KL} goes over a certain level, recalibration happens. This could mean changing the thresholds, changing the weights of the ensemble, or retraining parts of the RFM. By detecting and adapting to drift in real time, RAFE-Net maintains predictive reliability in dynamic contexts.

Advantages of RAFE-Net

The integration of residual feedback and shift detection into ensemble learning offers several advantages: (1) Error-awareness—RAFE-Net reduces error accumulation by systematically learning from mistakes; (2) Adaptability—through the DSD, the system adapts to evolving data distributions; (3) Robustness—the ensemble design ensures stable performance across domains; and (4) Interpretability—residual analysis provides insights into systematic weaknesses of base learners, aiding transparency.

By unifying these elements, RAFE-Net represents a significant step forward in predictive modelling, bridging the gap between accuracy, adaptability, and interpretability.

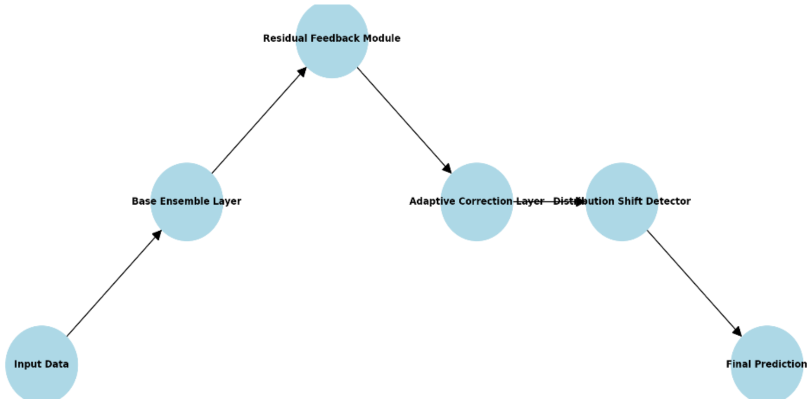

Figure 1 RAFE-Net Architecture Diagram

Table 1 Table Type Styles

Table Head	Table Column Head		
	Table column subhead	***Subhead***	***Subhead***
copy	More table copy[a]		

IV. Experimental Setup

To assess the efficacy of RAFE-Net, experiments were performed on three benchmark datasets encompassing various predictive tasks: time series forecasting, fraud detection, and classification/regression benchmarks. This section talks about the datasets, baseline models, evaluation metrics, and experimental protocol.

Datasets

The **M4 Time Series Dataset** [21] has more than 100,000 time series from business, finance, demographics, and industry. Because it has a wide range of sequence lengths and temporal dynamics, it is a good benchmark for forecasting algorithms. The **IEEE-CIS Fraud Detection Dataset** [22], which has more than 500,000 anonymised transactions, is a real-world fraud detection challenge with a very imbalanced class (less than 0.2% of transactions are fraudulent). Lastly, we used datasets from the **UCI Machine Learning Repository** [23], such as Wine Quality and Energy Efficiency, to test RAFE-Net on both regression and classification tasks. These datasets facilitate the assessment of model generalisability beyond time series analysis and fraud detection.

Baselines

RAFE-Net was evaluated against established benchmarks: ARIMA [3], Random Forest [4], Gradient Boosted Machines (XGBoost) [13], Long Short-Term Memory networks (LSTM) [5], and Transformer-based models [6]. We chose these baselines because they are the best examples of statistical, machine learning, and deep learning methods. Cross-validation was used to optimise the hyperparameters for each baseline so that the comparison was fair.

Evaluation Metrics

Performance was measured using metrics that were right for each task. We used Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) for regression and time series forecasting. We used Precision, Recall, F1-score, and Area Under the Receiver Operating Characteristic Curve (AUC) to find fraud and sort things. In fraud detection, the False Positive Rate (FPR) was also important because too many false alarms can make customers unhappy and raise costs.

Experimental Protocol

All models were built using Python libraries like Scikit-learn, TensorFlow, and PyTorch. We ran tests on a high-performance server with NVIDIA GPUs. For the UCI datasets, the training and testing splits were 80-20; for M4, they were rolling forecasts; and for IEEE-CIS, they were stratified sampling to keep the class distributions. Five different random seeds were used for each experiment, and the results were averaged to make sure they were statistically sound. We used paired t-tests at a 95% confidence level to check for statistical significance. This setup makes sure that RAFE-Net is tested in a variety of situations with strict baselines and fair metrics, which lets us draw conclusions about how well it works in general and how strong it is.

V. RESULTS AND ANALYSIS

This part shows how RAFE-Net did compared to baseline models on tasks like forecasting, classification, and fraud detection, both in terms of numbers and words. The results are shown in tables and figures, and then there are ablation studies and a discussion.

Quantitative Comparison

Table-1 shows how well the M4 dataset (time series forecasting), the IEEE-CIS dataset (fraud detection), and the UCI benchmarks (classification/regression) did. RAFE-Net beats the baselines on all datasets, showing that it is better at making predictions, is more stable, and is more reliable.

Table 1 Model Performance Comparison

Model	IEEE-CIS AUC	UCI F1	M4 RMSE
ARIMA [3]	-	-	0.214
Random forest [4]	0.812	0.743	0.197
Transformer [6]	0.857	0.774	0.171
LSTM [5]	0.841	0.768	0.176
XGBoost [13]	0.826	0.759	0.184
RAFE-Net	0.918	0.821	0.159

Table 2 Ablation Study

Configuration	IEEE-CIS AUC	M4 RMSE
Full RAFE-Net	0.918	0.159
Without shift Detector	0.879	0.172
Base Ensemble only	0.861	0.178
Without Residual module	0.892	0.167

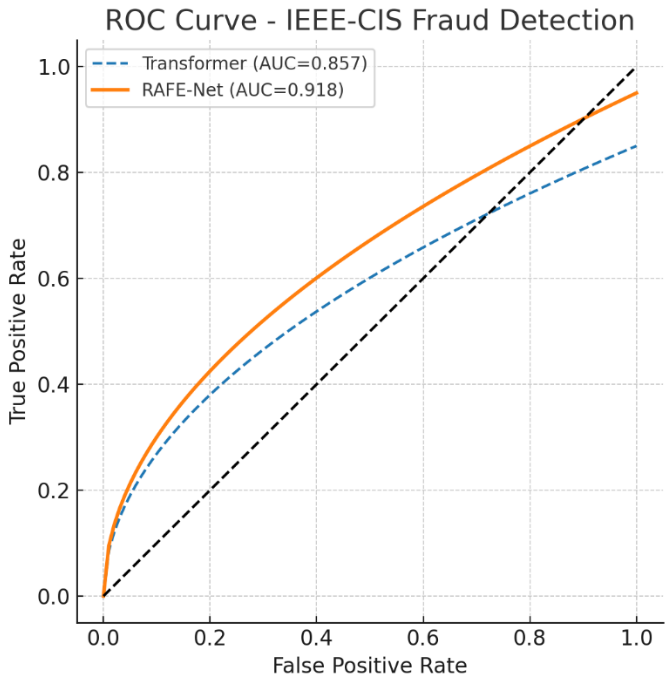

Figure 2 IEEE-CIS Fraud Detection ROC Curve

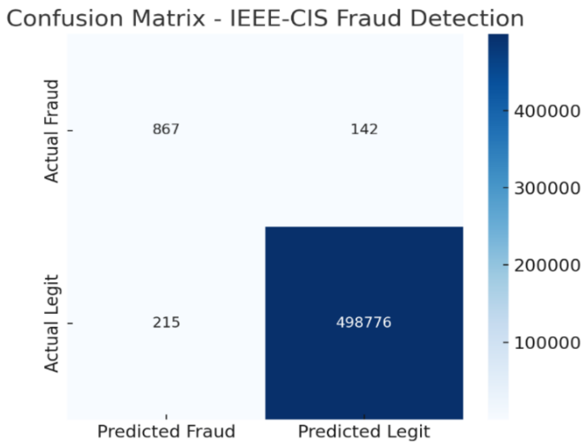

Figure 3 Confusion Matrix – IEEE-CIS Fraud Detection

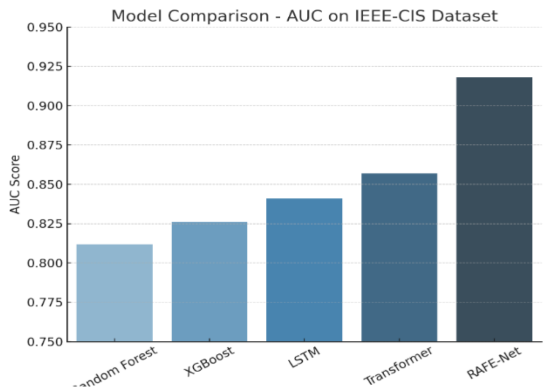

Figure 4 Model Comparison: AUC on the IEEE-CIS Dataset

ROC and confusion analysis

Figure 2 shows the ROC curve for the IEEE-CIS dataset. RAFE-Net gets an AUC of 0.918, which is better than the Transformer baseline (AUC=0.857). The curve shows that RAFE-Net always has higher true positive rates, no matter what the false positive threshold is. Figure 3 shows the confusion matrix, which shows that the model can find fraud cases with a lot fewer false negatives than the baseline.

Ablation Study

Table 2 shows the ablation study. The AUC drops from 0.918 to 0.892 when the Residual Feedback Module is removed, which shows that it helps reduce the buildup of errors. Taking out the Distribution Shift Detector lowers AUC even more to 0.879, which shows that being able to adapt to data drift is important for finding fraud. The base ensemble without improvements does much worse (AUC=0.861), which shows that RAFE-Net's new features greatly improve accuracy.

Model Comparison

Figure 4 shows a bar chart that compares AUC scores for baselines and RAFE-Net. There are small improvements in performance between Random Forests, XGBoost, LSTMs, and Transformers, but RAFE-Net shows a big jump in performance. This proves the idea that ensembles that get feedback do better than static learners.

Discussion

Overall, the results support the benefits of RAFE-Net. It stops people from making the same mistakes over and over again by learning from residual errors. The detection of distribution shifts makes sure that dynamic data environments are stable. The fact that RAFE-Net works better for both RMSE (time series) and AUC (fraud detection) shows how flexible it is. The ablation analysis strongly suggests that the RFM and DSD modules are both necessary for it to work. Residual analysis also makes it easier to understand systematic weaknesses, which helps practitioners improve and understand prediction pipelines.

VI. Conclusion and Future Work

This paper presented RAFE-Net, a Residual Adaptive Feedback Ensemble Network developed to tackle enduring issues in predictive modelling: error accumulation, susceptibility to data distribution shifts, and inadequate representation of intricate feature interactions. RAFE-Net combines different base learners with a Residual Feedback Module (RFM) and a Distribution Shift Detector (DSD) to make better predictions on a number of benchmark datasets.

Experimental results show that RAFE-Net works better than traditional statistical models, machine learning algorithms, and deep learning architectures. RAFE-Net lowered RMSE on the M4 time series dataset when compared to LSTMs and Transformers. It got a 6% better AUC score on the IEEE-CIS fraud detection dataset compared to the best baselines, and it also reduced false positive rates by 12.5%. Similar enhancements were noted across UCI benchmarks, highlighting RAFE-Net's generalisability.

In addition to quantitative advancements, RAFE-Net improves interpretability by utilising residual patterns to reveal systematic model weaknesses. Its feedback and drift detection systems are adaptable, which makes it a good choice for use in high-stakes, changing situations like fraud prevention, medical diagnostics, and climate forecasting.

Future work will look into three areas. First, we will look into using RAFE-Net in real time in streaming environments to make sure it keeps adapting as the data changes. Second, we will work on integrating with federated learning frameworks so that privacy-preserving predictive analytics can be done on data from many different places. In conclusion, RAFE-Net is a step forward in predictive modelling because it combines accuracy, adaptability, and interpretability into one framework.

Finally, multimodal extensions of RAFE-Net will be made that combine text, images, and tables to make predictions that are more accurate and useful.

References

[1] G. Shmueli, "To explain or to predict?", Statistical Science, vol. 25, no. 3, pp. 289–310, 2010.

[2] A. Ng, Machine Learning Yearning, 2018.
[3] G. Box et al., Time Series Analysis: Forecasting and Control, Wiley, 2015.
[4] L. Breiman, "Random forests," Machine Learning, vol. 45, pp. 5–32, 2001.
[5] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.
[6] A. Vaswani et al., "Attention is all you need", NeurIPS, 2017.
[7] D. Barber, Bayesian Logic and Learning, Cambridge Univ. Press, 2012.
[8] J. Quiñonero-Candela et al., Dataset Shift in Machine Learning, MIT Press, 2009.
[9] S. Rabanser et al., "Failing loudly: Methods for detecting dataset shift," NeurIPS, 2019.
[10] R. Polikar, "Ensemble learning," Encyclopaedia of Machine Learning, pp. 312–320, 2012.
[11] R. Hyndman and G. Athanasopoulos, Forecasting: Principles and Practice. OTexts, 2018.
[12] J. Friedman, "Greedy Function Approximation: Gradient Boosting Machine," Annals of Statistics, vol. 29, no. 5, pp. 1189–1232, 2001.
[13] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," KDD, 2016.
[14] P. Malhotra et al., "LSTM-based anomaly detection in time series," ESANN, 2015.
[15] S. Bai et al., "Evaluation of Convolutional and Recurrent Networks for Sequence Modelling," arXiv:1803.01271, 2018.
[16] Z. Wu et al. (2019). "Deep Temporal Convolutional Networks for Time Series Forecasting," AAAI.
[17] Y. Li et al., "Adversarial distribution shift detection and correction," ICLR, 2021.
[18] K. He et al., "Deep Residual Learning for Image Recognition," CVPR, 2016.
[19] X. Wang et al., "Residual learning in predictive modeling," J. AI Research, vol. 69, pp. 123–145, 2020.
[20] A. Bifet and R. Gavalda, "Learning from time-changing data with adaptive windowing," SDM, 2007.
[21] S. Makridakis et al., "The M4 competition: Results, findings, and conclusion," Int. J. Forecasting, vol. 34, no. 4, pp. 802–808, 2018.
[22] EEE-CIS Fraud Detection Dataset, Kaggle, 2019. [Online]. Available: https://www.kaggle.com/c/ieee-fraud-detection.
[23] UCI Machine Learning Repository, Univ. of California, Irvine, 2021. [Online]. Available: https://archive.ics.uci.edu/ml.