Optimizing Hyperparameters for Customer Churn Prediction with PSO-Enhanced Composite Deep Learning Techniques

Mohammad Sedighimanesh[1*], Ali Sedighimanesh[1], Hesam Zand Hesami [2]

[1]. Independent Researcher, Specialist in Artificial Intelligence and Data Analysis, Tehran, Iran
[2]. Department of Management and Economics, Science and Research branch, Islamic Azad University, Tehran, Iran

Abstract

For Telecom operators, customer churn, i.e., the event when the customers leave a service provider, becomes a critical concern, studies have shown that acquiring new customers cost five times more than to retain them. In competitive markets, where is increasingly important, to sustain growth as well as profitability correctly predicting the tendencies for customer churn is important. Traditional predictive fashions frequently underperform due to the complex nature of client behavior. In this examine, we introduce a unique composite deep mastering framework whose hyperparameters are optimized the usage of the Particle Swarm Optimization (PSO) set of rules. Our method integrates a couple of neural community architectures to effectively capture each spatial and temporal patterns in client interactions. The PSO set of rules systematically first-rate-tunes parameters including activation functions, regularization techniques, gaining knowledge of rates, optimizers, and neuron counts—ensuing in a model that demonstrates robust overall performance. We evaluated our approach the usage of key metrics consisting of accuracy, precision, recollect, F1 score, and ROC AUC on a numerous purchaser dataset. Comparative analyses were conducted in opposition to established deep studying fashions (LSRM_GRU, LSTM, GRU, CNN_LSTM) in addition to other conventional methods (KNN, XG_BOOST, DEEP BP-ANN, BiLSTM-CNN, and Decision Tree). Experimental results stompy that our PSO-enhanced composite deep learning model stands out significantly compared with conventional models. Comparing the ROC-AUC scores of 0.932 and 0.93, F1 scores of 0.90 and 0.895, and accuracy rates of 83.2% and 93% on both Cell2Cell and IBM Telco datasets. it is indeed effective for practical churn prediction use incitements efficiencies. Var The experimental results demonstrate that our PSO express tree model outperforms conventional methods, achieving better performance with ROC totter score above 0.932 and 0.93, F 1 scores above 0.90 and 0.895 as well as accuracy rates in excess of 83.2% (% paper) and 93% (on the Telco data set) for Cell2Cell and IBM Telco respectively. This is further confirmation of its effectiveness and promise for practical churn prediction applications.

Keywords: Customer Churn Prediction; Hyperparameter Optimization; Particle Swarm Optimization (PSO); Deep Learning Models; Telecommunications Analytics.

1- Introduction

The fast upward thrust of e-commerce platforms has transformed client engagement and retention techniques. In a aggressive panorama, predicting client churn is vital for commercial enterprise increase and profitability [1]. Although considerable studies have evolved various churn prediction fashions, the dynamic nature of purchaser conduct and records complexity, mainly in telecom with its substantial and diffused interaction information, pose ongoing challenges.

Advances in system mastering (ML) and deep gaining knowledge of (DL) provide promising answers, yet their effectiveness relies upon on precise hyperparameter tuning—a complicated mission because of high-dimensional seek spaces and computational costs[2] [3].

Despite the widespread use of ML and DL in the prediction of pimple, there is a remarkable difference in adapting hyperparameters to promote accuracy and efficiency. Traditional methods such as web searches and random discovery are computational and often sub -form [4]. General DL technology, which utilizes diverse nervous network strength, and complicates further setting.

✉ **Mohammad Sedighmanesh**
mohammad.sedighimanesh@gmail.com ,

This study addresses this gap by suggesting a bio-induced algorithm, particle crew optimization (PSO) to adapt to hyperparameters in a composite DL model for telecommunications spread [5].

This research goals to broaden and validate a powerful approach for optimizing hyperparameters in composite deep mastering fashions for telecom churn prediction. It pursues 3 targets:

- Create a PSO-embedded composite DL framework integrating Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, and Gated Recurrent Units (GRUs) to capture nuanced client behavior and transaction styles, superior by way of a novel integration of PSO with dropout regularization and mini-batch schooling for robustness and efficiency.
- Apply Particle Swarm Optimization (PSO) to song hyperparameters, enhancing churn prediction accuracy and computational performance over conventional techniques like random seek and grid search.
- Evaluate the PSO-enhanced version's overall performance in opposition to conventional models (LSRM_GRU, LSTM, GRU, CNN_LSTM) the usage of metrics along with accuracy, precision, consider, F1 rating, and ROC AUC.

The difference identified by achieving these goals will be addressed, a new approach will be offered to improve the grinding prediction model and telecommunications companies will be given action - provoking insights to improve retention strategies. This research contributes significantly to the future analysis and telecommunications spread:

- Novel PSO integration: We suggest a creative method that uses PSO for fine-tuning hyperparameters in the general DL model, improves the efficiency and performance of traditional techniques, as validation is valid.
- This study makes widespread comparisons with models such as LSTM, GRU and CNN_, and perform better performance in accuracy, precision, recall, F1 points and ROC AUC, which promotes BI-inspired hyperparameters.
- Practical framework: It provides a scalable, adaptable PSO-DL model for telecom and improves the prediction in different data sets.
- And analysis goals: It shows PSO's ability to adapt to DL models, and encourages the further discovery of the evolutionary algorithm in large data and complex predictions.

The paper proceeds as follows: Section 2 opinions existing work on churn prediction and optimization strategies. Section 3 describes the PSO algorithm and its integration into our model. Section four outlines the proposed technique, which include information training and version improvement. Section five offers the experimental results, observed by a discussion in Section 6 and conclusions in Section 7.

2- Related Work

Understanding and predicting customer churn has been a focal point of research across various sectors, particularly the telecom industry, where the possibility of accurately predicting churn has far reaching implications in terms of business revenue and growth. Historically, churn prediction models primarily rested on statistical and machine learning methods, such as logistic regression, decision trees, and support vector machines (SVMs)[6]. Although effective to a degree, these methods often lacked the ability to identify complex and non-linear patterns hidden within the large volumes of customer interactions and behaviors. The ever-increasing complexity of these patterns has forced the exploration of more sophisticated analytical techniques that are capable of accurately identifying and predicting customer churn [7].

It has been proved that recent progress in deep learning yields productive ways for improving churn prediction models. Deep learning is powerful because it is able to learn hierarchical data representations and, consequently, it has been shown to noticeably outperform traditional machine learning models in identifying complex patterns in large datasets[8]. Various deep learning architectures, e.g., Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs) and their flavors like Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs) have been tentatively applied to churn prediction and these models, have displayed state-of-the-art performance on myriad sequential and time-series data – making them tailor-made for analyzing customer interaction sequences, and transaction histories [9].

The performance of deep learning models depends heavily on the choice of hyperparameters. Manually tuning these parameters is an impossibly time-consuming process, which is why automated hyperparameter optimization techniques like Grid Search, Random Search, Bayesian Optimization, and Evolutionary algorithms such as Genetic Algorithms (GAs) and Particle Swarm Optimization (PSO) have become extremely popular. Of these, PSO, a bio-inspired optimization algorithm, has shown promise in navigating the hyperparameter search space efficiently [10]. The objective of optimizing a given objective function with respect to a model's hyperparameters is universal in the field of machine learning. This objective is commonly addressed using grid, random or other more computational expensive techniques. Particle Swarm Optimization (PSO) provides a heuristically driven solution to this problem, simulating the social behaviors of bird flocking or fish schooling. Applications for this recently have been numerous within various domains. However, PSO application to optimizing hyperparameters for deep learning-based churn prediction models is underexplored[11].The individual components of churn prediction, deep learning and hyperparameter optimization

have been thoroughly investigated in the literature. However, there is a dearth of studies within the research literature that investigate the integrated application of PSO to tune composite deep learning models in churn prediction. This present void in the research literature presents an enticing prospect of making substantial contributions to this field of research. We are proposing a novel hybridization of PSO algorithm against a deep learning neural network architecture to attempt to simultaneously address the problems of churn prediction that have surpassed the capabilities of current practices.

We discuss five key innovations in the evolving landscape of churn prediction using the Deep Learning (DL) and Machine Learning (ML)[12], [13]: the rise of DL, the creation of embedded space using DNN, the advent of TL and Ensemble-based Meta-Classification, proposed in "TL-DeepE", Associative Chaos Networks, a radically new DNN architecture, that was introduced in, Jazz Networks, which presented Chaos and Laws through the lens of music, Cloud Machine Learning in Healthcare by Adam Gasiewski and Mark W. Hester and Epsilon Computing's ETL as a Service. This method achieved high accuracy on telecom datasets — underscoring the synergies between TL and ensemble methods in churn prediction — by fusing fine-tuned DCNNs with ensemble methods.

Subsequent to its successful use of DCNN in modeling data at successive levels of abstraction, follow-up work saw such an architecture (ANN) outperform traditional models on the IBM Telco Churn dataset, thus demonstrating that artificial neural networks such as these, which feature self-learning capabilities, are able to readily outperform conventional algorithms by automatically and efficiently learning from big data [7].

In addition to these studies, a study used a novel model to evaluate the performance of a Customer Churn Prediction (CCP) model with a Naïve Bayes classifier shedding light on the performance of different sample sets, suggesting feature selection methods as an area for future work. Another study applied a Deep Learning method to a Telco dataset, yielding high-accuracy churn prediction by analyzing customer attributes, underscoring DL as a useful tool for identifying the key churn predictors. Finally, exhaustive analyses with various ML algorithms have arrived at the optimal churn prediction classifiers, chief among them being Logistic Regression. The collective work they did here is important as this not only advances our understanding of churn prediction mechanisms, it also opens new doors for the application of DL and ML to provide deep insights and drive actions in customer retention strategies.

The study [3], is a detailed description of how to forecast customer churn, a critical area for telecoms that are looking to keep and grow their subscriber base. From a study that "introduces a comprehensive approach for churn prediction in the telecommunications industry based on Deep Learning (DL) and Machine Learning (ML) with independent evaluations of both the activation functions and two feature selection methods," the paper details how the authors developed a predictive model using Deep Backpropagation Artificial Neural Networks (Deep-BP-ANN) integrating it with two different feature selection methods:

Various techniques employed in this research include Variance Thresholding and Lasso Regression and this model is further refined with the use of an early stopping technique that prevents overfitting, which is a common problem with Machine Learning models. The authors employ dropout and activity regularization to minimize overfitting. The performance of the model is evaluated using both holdout and 10-fold cross-validation evaluation methods. Random Oversampling is also used to balance the dataset, since in real world customer churn datasets, there are comparatively much smaller number of churners. The results show that Deep-BP-ANN model perform better and the combination of lasso regression for feature selection and activity regularization perform exceedingly well in predicating customer churn, outperforming traditional Machine Learning (ML) techniques like XG_Boost, Logistic Regression, Naïve Bayes, and KNN. This performance is consistent across two real-world telecom datasets: IBM Telco and Cell2Cell.

The paper [3] points out that the Deep-BP-ANN model improved churn prediction accuracy by over 17%, against other deep learning models and over 22%, against other ML techniques on the same datasets. The use of lasso regression for feature selection, and early stopping, to find the optimal number of the epochs were central to why the model works so well. The findings suggest that deep learning may be a highly inclusive and therefore, low-cost method due to cheaper filtered data, to process completced feature relationships in complex, large scale", churn prediction works. The study also identifies some caveats, including the use of datasets created for the purpose (DS and IC) that". may not capture many established challenges of the telecom industry".

The study [4] looks at the modern organizations, where customer tend to switch over to competitors due to poor service quality and satisfaction. It introduces a novel deep learning model, BiLSTM-CNN, that predicts customer churn with far greater accuracy than the more traditional machine learning models. The abstract sets the stage for the research by pointing out that existing ML/DL algorithms have many limitations when it comes to customer churn prediction, as they fail to forecast accurately. By integrating Bidirectional Long Short-Term Memory (BiLSTM) and Convolutional Neural Networks (CNN), the model aims to effectively capture and analyze the customer data to foresee churn at an accuracy of 81%, as demonstrated in a benchmark dataset.

The paper shows that the proposed BiLSTM-CNN outperforms several traditional machines learning classifiers such as Support Vector Machines, Decision

Trees, K-Nearest Neighbors in predicting customer churn [4]. This advantage comes from its ability to take into account sequential data in both ways, and then capturing patterns of customer behavior more extensively. The paper then evaluates the precision, recall and F1 scores of the model, and compares them in detail with similar metrics of existing machine learning models as even well as deep learning models. Finally, the paper underscores the model's effectiveness in increasing the accuracy of churn prediction, which can be critical for telecom companies in deploying ad-hoc customer retention strategies. The paper also lists a number of limitations - some of which suggest avenues for future work, like the focus on binary classification and reliance only on numerical features, which could be the target of multi-dimensional CNN approaches; others touch upon broader possibilities, such as the use of multiclass classification to reduce feature zipping, and incorporation of a wider range of features for more precise predictions.

The paper [5] examines the very important problem of customer churn within the telecommunications sector, and the development of a predictive model that utilizes both parsing and clustering techniques. This model comes into being with the hopes of identifying customers likely to churn and why, in a telecom industry with vast amounts of data that is produced daily. Using a feature selection process with information gain and correlation attribute evaluation filters, our approach is able to successfully classify customer data using random forest, with 88.63% accuracy for correctly classified instances. In the next step, we performed a post processing approach on the churned customers by using Cosine Similarity to segment them into clusters that would help in retention owed to specific behavior and preference of the customers.

This approach has proven help to achieve a high accuracy in churn prediction while being able to help customer carriers to distinguish which customers are most likely to leave their network provider. The identification of causes of churn coming from low-level application data has the possibility to practical afford operators to direct their marketing campaigns as well as subscriber offerings for this activation. This would lead mobile carriers develop marketing and retention campaigns which are specifically designed for its subscribers, while being able to make quite sure. Instead of using the information which one thinks will attract those who are most likely to leave. The study acknowledges some limitations, including the model's dependency on particular datasets, and propose several research directions for enhancing the model's applicability to diverse datasets, as well as the integration of further predictive techniques. This paper is based on work that has been funded by a variety of research funding. The extensive teamwork behind this research has played a key role in the advancement of churn prediction methodologies in the telecom sector.

The study [14] assesses the importance of hyperparameter tuning in improving the performance of deep learning models used for predicting customer churn in the telecommunications sector. The abstract suggests that the focus of the paper will be on comparing multiple machine learning techniques while giving "special attention on deep learning" for the purpose of predicting churn of customers. Furthermore, it states that there are very few empirical studies that show how hyperparameters influence the model's performance. The authors experiment with the different configuration including: type of optimizers, activation functions and batch sizes and argue that using ReLU in the hidden layer and the sigmoid function in the output layer provides the best accuracies of the model in predicting churn.

Unsurprisingly, the results show that the model's performance is considerably greater when the ReLU activation function in hidden layers is used in conjunction with the sigmoid function in the output layer. Due to this configuration the model has an accuracy of close to 86.9%. It was also noted that using smaller batch sizes can actually be better for the overall performance of the model with a noticeable drop in its performance as the batch sizes approached the size of the test dataset. A deeper look at the different optimizers also found that the RMSProp optimizer outperformed the others over the 500 epochs proving its ability to reduce the loss function and increase the predictive accuracy of the model. This led the conclusion that the hyperparameter tuning is a critical component of any deep learning models for churn prediction and that the right combination of activation functions, batch sizes and optimizers could greatly increase performance. This study is important as it advances both the theoretical and practical understanding of churn prediction within the telecommunications sector and will serve as a guide for researchers to continue to improve deep learning models for this application [14].

The study [15] focuses on enhancing patron retention techniques inside the telecommunication industry with the aid of predicting patron churn the usage of machine studying strategies. The observe proposes using a Multilayer Perceptron (MLP) neural network and compares its performance with traditional statistical techniques together with Multiple Regression Analysis and Logistic Regression Analysis. The effects indicate that the neural network achieves a advanced accuracy of 91.28% in predicting consumer churn, significantly outperforming the regression-based totally models. This shows that system mastering models, mainly neural networks, provide a extra powerful technique to churn prediction and may be used to decorate patron retention efforts.

The paper's key contributions include the application of a synthetic neural community to the client churn trouble, demonstrating its blessings over conventional statistical strategies. The research methodology includes records

extraction, preprocessing, and the software of MLP neural networks, with overall performance measured based totally on accuracy, sensitivity, and specificity. A fundamental hindrance of the take a look at is the reliance on a limited dataset from a unmarried telecommunication issuer, which may affect generalizability. Nevertheless, the findings fortify the effectiveness of neural networks in predictive analytics, imparting a precious opportunity to traditional statistical techniques for improving client retention techniques in competitive markets.

The research paper [16] addresses customer churn, a major concern for telecom companies. It proposes a predictive model to identify customers likely to leave using machine learning techniques. The study applies the enhanced Relief-F feature selection algorithm to refine the dataset and employs Random Forest and Convolutional Neural Networks (CNN) for classification. Results show CNN achieving a 94% prediction accuracy, outperforming Random Forest at 91%. The study highlights the importance of feature selection in improving prediction accuracy and suggests that CNN is the superior model for churn prediction in telecom.

The methodology leverages statistics mining and device getting to know, especially Relief-F for characteristic choice, accompanied via category the use of Random Forest and CNN. The look at makes use of a dataset of three,333 telecom clients, selecting 14 key functions for analysis. The research demonstrates that CNN's deep getting to know method affords extra particular churn predictions than traditional system mastering models. However, the observe acknowledges obstacles, such as the need for in addition optimization and exploration of other gadget getting to know fashions. The conclusion emphasizes that telecom corporations can notably enhance client retention strategies by way of adopting superior AI strategies like CNN, ensuring greater powerful churn prediction and commercial enterprise growth.

The paper [17] explores the position of machine getting to know in predicting patron churn and enhancing retention techniques inside the telecom industry. It discusses conventional churn prediction challenges, inclusive of information fine troubles and confined version effectiveness, and highlights the benefits of gadget mastering strategies like selection trees, guide vector machines, and ensemble techniques. The paper emphasizes how device getting to know allows actual-time statistics analysis, improves scalability, and enhances predictive accuracy. It additionally identifies ethical issues related to data privateness and the want for interpretable AI fashions. The research suggests that integrating AI-pushed predictive analytics can extensively reduce churn, optimize retention strategies, and enhance telecom commercial enterprise overall performance. The study contributes by analyzing

gadget gaining knowledge of fashions against traditional statistical techniques, proving the superiority of AI-based totally processes for churn prediction. Its technique includes reviewing existing literature, comparing exclusive gadget learning algorithms, and discussing demanding situations such as overfitting, model selection, and implementation charges. The paper recognizes obstacles, which include records integration problems and the want for interpretable AI solutions. The findings recommend that actual-time prediction, personalized retention techniques, and AI-pushed customer support automation will shape the destiny of telecom purchaser management. The conclusion underscores the importance of machine getting to know in reducing churn and improving purchaser loyalty, making AI-driven retention techniques essential for telecom companies' long-term success.

The study [18] explores the task of purchaser churn within the telecommunications area and the effectiveness of gadget studying fashions in predicting it. The observe evaluates diverse classifiers, which includes Random Forest, XGBoost, LGBM, Logistic Regression, Decision Trees, and an Artificial Neural Network (ANN). It employs characteristic selection, hyperparameter tuning, and ensemble averaging to optimize overall performance. The results display that the LGBM and XGBoost fashions outperform others, with the best accuracy of eighty.36%. The research highlights the significance of gadget studying in enhancing consumer retention charges and operational performance in telecom groups. This look at contributes via evaluating multiple machines getting to know models and showcasing the advantages of ensemble methods in churn prediction. It follows a structured technique, which includes information preprocessing, exploratory data analysis, model education, and assessment the use of cross-validation strategies. The number one hassle of the take a look at is its reliance on a publicly available dataset as opposed to real-world telecom statistics, which may affect generalizability. The findings recommend that machine getting to know fashions, particularly ensemble procedures, provide more accurate churn predictions than conventional techniques. The conclusion emphasizes the need for telecom groups to adopt superior predictive analytics and suggests destiny research into integrating blockchain-based totally solutions for secure consumer statistics management.

Table 1: Comparison of algorithms

Category	Abstract	Contributions	Methods Used	Results	Conclusions	Limitations
"Customer Churn Prediction in Telecommunication Industry Using Deep Learning" [3]	Explores Deep Backpropagation ANN with feature selection for churn prediction.	Demonstrates DL models' efficacy in churn prediction with optimized feature selection.	Deep-BP-ANN, Variance Thresholding, Lasso Regression.	Achieved high accuracy on telecom datasets, outperforming traditional ML methods.	Validates the potential of DL for churn prediction with appropriate feature selection.	Limited by specific datasets; may not generalize across the telecom sector.
"Customer Churn Prediction Using Composite Deep Learning Technique" [4]	Introduces a novel BiLSTM-CNN model to enhance churn prediction accuracy.	Shows BiLSTM-CNN model's superior accuracy over traditional ML methods.	BiLSTM-CNN.	Reached 81% accuracy, surpassing conventional classifiers.	Confirms the BiLSTM-CNN model as an effective tool for telecom churn prediction.	Focused on binary classification and numerical features only.
"A Churn Prediction Model using Random Forest: Analysis of ML Techniques for Churn Prediction" [5]	Develops a churn prediction model combining classification and clustering via Random Forest.	Highlights the effectiveness of integrating classification and clustering for detailed churn analysis.	Random Forest, information gain, correlation attribute ranking.	Random Forest model showed high accuracy and provided insights into churn reasons.	Proves the utility of combining methods for a nuanced understanding of churn.	Model's dependence on specific datasets could limit broader applicability.
"Impact of Hyperparameters on Deep Learning Model for Customer Churn Prediction in Telecommunication Sector" [14]	Investigates the impact of hyperparameter tuning on deep learning model performance for churn prediction.	Emphasizes the significant role of hyperparameter tuning in improving model performance.	Activation functions, batch sizes, optimizers in DL models.	Found optimal combinations of activation functions and optimizers that significantly improved accuracy.	Highlights the critical impact of hyperparameter tuning on churn prediction models.	Study's reliance on a synthetic dataset may not fully represent real-world complexities.
A Multi-Layer Perceptron Approach for Customer Churn Prediction[15]	The look at explores the software of a Multilayer Perceptron (MLP) neural network to expect consumer churn in the telecommunication zone, comparing it with conventional statistical models like Multiple and Logistic Regression.	Introduces MLP as a superior predictive device for consumer churn, demonstrating its higher accuracy (91.28%) as compared to statistical procedures. Provides insights into how telecom providers can proactively maintain customers.	Data series from a Malaysian telecom employer, preprocessing, function extraction, and evaluation using MLP neural network and statistical fashions (Multiple Regression and Logistic Regression).	MLP neural community carried out the highest accuracy (91.28%) in predicting patron churn, outperforming Multiple Regression (78. 84%) and Logistic Regression (75.19%).	The study confirms the superiority of the nervous network on the traditional statistical model to predict the customer's brainstorming, and exposes their ability to increase customers' storage strategies.	Requires excessive computational power, capability overfitting of the version, and the need for non-stop updates with new purchaser statistics for foremost
A Churn Prediction System for Telecommunication Company Using Random Forest and Convolution Neural Network Algorithms [16]	The take a look at proposes a churn prediction version for telecom agencies the use of Random Forest and Convolutional Neural Network (CNN) classifiers. It objectives to enhance predictive accuracy by leveraging an stepped forward Relief-F function choice algorithm.	The quarter introduces a hybrid approach by combining random forest and CNN for prediction. Traditional methods show the effectiveness of deep learning in telecommunications analysis with better performance from traditional methods.	Data collection from a telecom dataset, characteristic extraction the usage of the Relief-F set of rules, and class the usage of Random Forest and CNN fashions.	CNN performed a better prediction accuracy (٩٤%) as compared to Random Forest ٩١%), demonstrating the capability of deep getting to know in churn prediction.	The study exposes CNN as a better method of predicting customer driving, and strengthens the need for advanced machine learning techniques in telecom analysis.	Computationally extensive fashions, requirement for large datasets, and demanding situations in actual-time implementation due to processing constraints.
implementing machine learning techniques for customer retention and churn prediction in telecommunications [17]	The paper explores the application of machine getting to know techniques in predicting patron churn and enhancing retention in telecommunications. It evaluates diverse gadget learning fashions, discusses data-related demanding situations,	Highlights the advantages of machine learning over traditional churn prediction methods. The author presents three potential improvements which include real-time analytics together with explainable AI systems and	Analysis of decision bushes, guide vector machines, ensemble gaining knowledge of, and deep studying fashions. Comparison of system gaining knowledge of techniques with conventional statistical	The performance of random forests and gradients as a dress algorithm surpasses traditional methods in identifying thought-to-be ill-formed notions. The teaching methods that deliver intensive instruction led to additional learning gains	Machine gaining knowledge of presents superior scalability, accuracy, and real-time processing for churn prediction. The have a look at emphasizes integrating AI-pushed retention techniques for telecom companies	Challenges in information availability and excellent, computational necessities, interpretability of deep mastering fashions, and moral issues in managing purchaser statistics.

Category	Abstract	Contributions	Methods Used	Results	Conclusions	Limitations
	and shows innovations for improving predictive accuracy.	customized retention approaches.	techniques for churn prediction.	although they do not explain concepts.	to decorate patron loyalty.	
Customer Churn Prediction for Telecommunication Companies using Machine Learning and Ensemble Methods[18]	The study checks the customer who throws himself in the telecom sector using the machine learning classification, including Random Forest, XGBOST, LGBM, Logistic Region and Decision Tree. The study forecasts use learning to improve accuracy and customers' storage strategies.	Introduces an optimized ensemble model for churn prediction, highlighting the effectiveness of Random Forest, XGBoost, and LGBM. Provides a comparative evaluation of multiple classifiers and hyperparameter tuning techniques.	Utilizes a telecom churn dataset, preprocessing strategies, and device studying classifiers (Random Forest, XGBoost, LGBM, Logistic Regression, Decision Trees, and ANN). Hyperparameter tuning and ensemble averaging had been applied for optimization.	LGBM and XGBoost done the best accuracy (80%) amongst examined models. The ANN model reached 79% accuracy, barely lower than the ensemble strategies. The take a look at confirms the effectiveness of ensemble learning in churn prediction	Machine learning models, specifically ensemble strategies, offer advanced predictive performance for telecom churn prediction. Optimized fashions can beautify consumer retention techniques for telecom corporations.	A specific dataset, which limits questions of potential generality, calculation complexity of the artists' methods and limited to challenges in implementing real - time.

3- Background and Explanation

Particle Swarm Optimization (PSO)[19] is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. PSO simulates the social behavior of birds within a flock or fish within a school. Originally introduced by Kennedy and Eberhart in 1995, PSO is inspired by the social behavior patterns of organisms that move in groups. Unlike evolutionary algorithms, PSO is guided not only by the best solution (or position) found by the swarm but also by the best solution found by each individual particle.

PSO operates by initializing a group of random particles (solutions) and then searching for optima by updating generations. In every iteration, each particle updates its velocity and position based on two "best" values[20]:

- Pbest (Personal Best): The best solution (position) it has achieved so far. This value is updated if the current position is better than Pbest.
- Gbest (Global Best): The best solution found by any particle in the population. This value is shared and updated across all particles in the swarm.

The PSO formula to update the velocity and position of particles can be broken down into the following[21]:

- Velocity Update: The velocity of each particle is recalculated based on its Pbest and Gbest. The velocity update is influenced by cognitive and social components, where the cognitive component reflects a particle's own experience and the social component is the learning from the swarm.

- Position Update: The position is then updated based on the new velocity. This has the effect of each particle moving toward its Pbest and Gbest locations in every dimension of the search space.

PSO in Hyperparameter Optimization: When it comes to optimizing hyperparameters for deep learning models, PSO can be employed to traverse the hyperparameter space automatically and efficiently. Each particle represents a potential set of hyperparameters. The fitness of each particle is determined based on the performance of the deep learning model (e.g., accuracy, F1 score) trained with these hyperparameters. The swarm iterates and converges upon the best solution — the set of hyperparameters that enhances the model's performance. The strength of PSO lies in its simplicity, ease of implementation, and the fact that it can quickly converge to a good solution in complex and high-dimensional optimization problems without requiring gradient information, which makes it particularly ideal for complex and high-dimensional optimization problems such as hyperparameter tuning for deep learning models.

Significance in Churn Prediction: The application of PSO to optimize hyperparameters in composite deep learning techniques for churn prediction presents a groundbreaking utilization of swarm intelligence that enhances the accuracy and efficiency of the model, and offers a powerful means of addressing the time-consuming and often impractical manual hyperparameter tuning that faces the intractably vast search space involved in advancing the capabilities of predictive analytics in the arena of customer churn management.

Pseudocode for Particle Swarm Optimization (PSO)
1. Initialize the swarm of particles with random positions and velocities in the D-dimensional problem space.
2. For each particle, evaluate the fitness of its current position.

3. Set Pbest to the initial position of each particle.
4. Identify the particle with the best fitness and set Gbest to this particle's position.
5. While the termination criterion is not met (e.g., maximum number of iterations or a satisfactory fitness level):
   a. For each particle i in the swarm:
      i. Update the velocity based on Pbest and Gbest using the formula:
      V[i][d] = w * V[i][d] + c1 * rand() * (Pbest[i][d] - X[i][d]) + c2 * Rand() * (Gbest[d] - X[i][d])
      Where:
      - V[i][d] is the velocity of particle i in dimension d.
      - X[i][d] is the current position of particle i in dimension d.
      - Pbest[i][d] is the best known position of particle i in dimension d.
      - Gbest[d] is the best known position among all particles in dimension d.
      - w is the inertia weight.
      - c1 and c2 are cognitive and social parameters, respectively.
      - rand() and Rand() are random functions in the range [0,1].
      ii. Update the position of the particle using the formula:
      X[i][d] = X[i][d] + V[i][d]
      iii. Evaluate the fitness of the new position.
      iv. If the fitness of the new position is better than the fitness of Pbest[i], update Pbest[i] to the new position.
   b. Identify the particle with the best fitness among all Pbest positions and update Gbest if necessary.
6. Return Gbest as the best solution found.

Velocity Update Formula[20], [21]:
$$V_{i,d}^{(t+1)} = w \cdot V_{i,d}^{(t)} + c_1 \cdot \text{rand}() \cdot \left(\text{Pbest}_{i,d} - X_{i,d}^{(t)} \right) + c_2 \cdot \text{Rand}() \cdot \left(\text{Gbest}_d - X_{i,d}^{(t)} \right) \qquad (1)$$

Position Update Formula[20], [21]:
$$X_{i,d}^{(t+1)} = X_{i,d}^{(t)} + V_{i,d}^{(t+1)} \qquad (2)$$

Where:
- $V_{i,d}^{(t)}$ is the velocity of particle i in dimension d at time t.
- $X_{i,d}^{(t)}$ is the position of particle i in dimension d at time t.
- Pbest $_{i,d}$ and Gbest $_d$ are the best personal and global positions encountered so far.
- w is the inertia weight that controls the impact of the previous velocity on the current velocity.
- c_1 and c_2 are acceleration coefficients that control the personal and social contribution to the velocity update.

rand () and Rand () are two random functions generating numbers between 0 and 1, providing stochastic elements to the search.

4- Proposed Model

We introduce the model we have developed to predict Customer Churn. The model has four main stages as displayed in Figure 1. First, we introduce the model and talk through its data flow. We're building a model to predict customer churn; our improved model will take advantage of advanced deep learning techniques to capture intricate patterns in telecom customer data. We know that developing and implementing a model to predict churn is a meticulous process from data prep to model development and testing. Typically, the first step of such a process is to import necessary libraries for data handling, preprocessing, and model building. After that, a dataset is imported; in this project, we read our dataset, which was stored in a CSV file,

and performed Exploratory Data Analysis (EDA) which entails educating oneself about the characteristics of the dataset, identifying any missing values, outliers, or obvious patterns in the data, in an attempt to help the reader to better understand the dataset we are working with as a model is developed. This process is necessary because it allows us to understand the structure of the dataset and subsequently prepare it for the heavy lifting of the model building. We present the general process and flow of the system, including why it's needed and where the model fits in as a production system.

The next step involves using a more advanced pipeline for data preprocessing that leverages StandardScaler for normalizing the numerical features, and OneHotEncoder for encoding categorical features. This is extremely important in making the data compatible with the neural network model, and in ensuring all features have the same scale. Following that, the innovative neural network architecture is discussed. It is based on a combination of Convolutional Neural Networks (CNN) followed by a few Gated Recurrent Units (GRU) and Long Short-Term Memory (LSTM) layers, which allows it to learn both spatial and temporal patterns in the data. To optimize this architecture further, we made use of a function that uses evolutionary algorithms — specifically Particle Swarm Optimization (PSO) — for hyperparameter tuning that was wrapped within a scikit-learn Pipeline, making it as easy to use as the ones provided by the library itself. This function carefully evolves the model to identify a set of hyperparameters that provide the best predictive performance we could achieve in this task. Model evaluation is done in terms of accuracy, precision, recall, F1 score, and ROC AUC score. We show the model was able to achieve par performance in the training set, and very close to that in the testing set as well. The results are analyzed in depth and visualized. This workflow provides a strong demonstration of how deep learning and evolutionary algorithms can be

combined for the nuanced task of customer churn prediction, and provides significant contributions to both the field and to telecom companies who are looking to enhance their customer retention strategy.

This proposed approach, unique in the rigor of its methodology and in the elegant application of deep learning techniques combined with evolutionary algorithms, aims to set a new state-of-the-art in the context of churn prediction. It demonstrates a profound appreciation for the complexity in customer data and illustrates the ability of our model to navigate this complexity to make accurate churn predictions. As such, this research contribution provides not only a novel framework for the academic community, but also a useful model for practitioners in the domain of telecommunications who may wish to employ this model in a "real world" scenario.
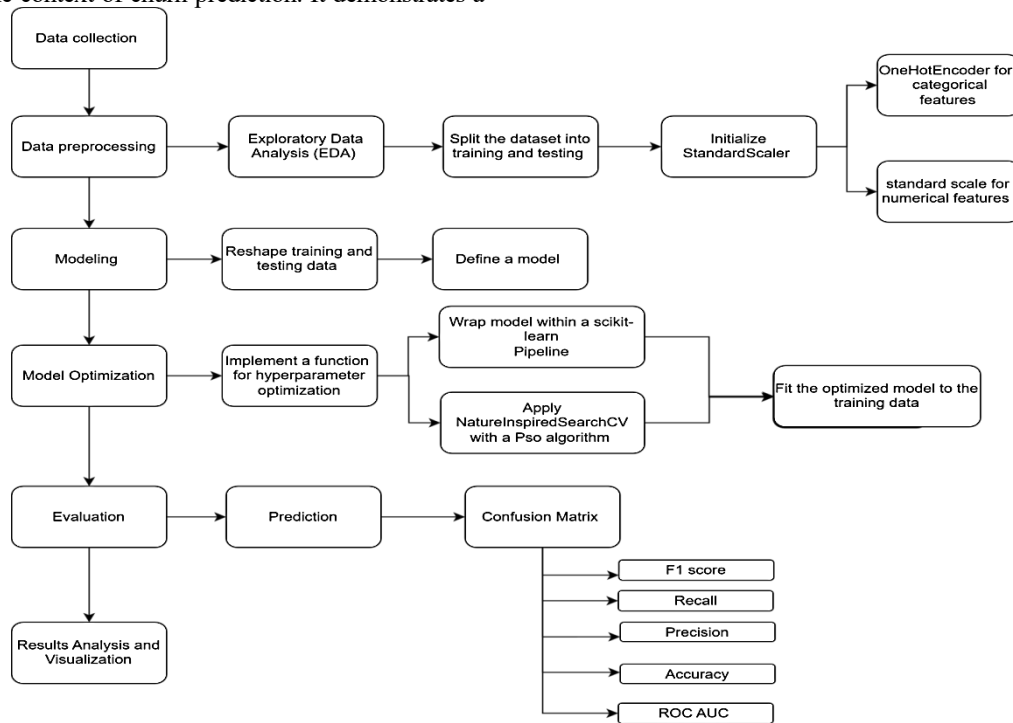

Fig. 1 Proposed system processes. Researcher's reference

The following section provides a concise pseudocode representation of the entire model development process

Pseudocode for Data Preparation and Model Development Process
A. Data Preparation and Model Development Process:
Import necessary libraries for data handling (pandas, numpy), preprocessing (scikit-learn), and model building (keras, tensorflow).
Load dataset from 'dataset.csv' into a DataFrame, ensuring integrity and accessibility.
B. Data Preprocessing:
Conduct Exploratory Data Analysis (EDA) to scrutinize dataset characteristics, missing values, outliers, and discern patterns.
Split dataset into 'train_set' and 'test_set' for unbiased model evaluation.
Initialize preprocessing utilities:
a. 'StandardScaler' for normalization of numerical features.
b. 'OneHotEncoder' for encoding categorical features.
Categorize features:
a. Categorical features for 'OneHotEncoder'.
b. Numerical features for 'StandardScaler'.
Apply 'ColumnTransformer':
a. Fit on 'train_set' and transform both 'train_set' and 'test_set'.
C. Model Preparation:
Reshape 'train_set' and 'test_set' to conform with neural network input structure.
Define neural network with CNN, GRU, and LSTM layers to capture data patterns.
Wrap model in 'KerasClassifier' for compatibility with scikit-learn.
D. Model Optimization:
Implement PSO-based optimization function:

---

a. Embed model within a scikit-learn 'Pipeline'.
b. Apply 'NatureInspiredSearchCV' with PSO for hyperparameter tuning.
Train model on 'train_set', ensuring robustness and generalizability.
E. Model Evaluation:
Assess model with 'train_set' and 'test_set' using metrics: accuracy, precision, recall, F1, and ROC AUC.
Document optimal hyperparameters post-optimization.
F. Results Analysis and Visualization

4-1- Dataset

This study utilizes two diverse datasets supported by the ability to perform more extensive analysis on churn prediction in the telecom industry. The first one is Cell2Cell and is often used in CRM research as a model. Its dataset contained about 71,047 instances, with each one represented by 58 different attributes describing various aspects of user interaction, service consumption, and personal identification. The data was provided by the Teradata Center for Customer Relationship Management at Duke University. This allowed the research to reach a deeper understanding of user interactions and fields where retention strategies might be applicable, such as the high-competitive telecom sector. [22], [23].

The second utilized dataset is our modified version of a public IBM Telco Customer Churn dataset that we changed by adding more fields and by modifications to make it closer to real. This dataset covers a wide variety of fields regarding customer attributes for a telecommunication company and allows us to make a reasonable concern of a relationship between customer actions and churn. Our version of a public "JB Link Customer Churn Problem" use-case covers the story of a budding California-based telecom provider, present in more than 1000 cities in 1600 zip codes. Even though JB Link has shown fast growth with a vibrant sales team that signed up many customers, in the past quarter, only 43% of new customers were retained. [22], [23].

Collected data, which was a random sample of 7,043 customers would be an invaluable source of information for our data science team. Not only would it help to deconstruct the driving factors behind the high churn rate but also form the basis of the machine-learning model, which accurately predicts which customers are likely to churn. The latter would, in turn, provide the basis for an individualized retention strategy, which would greatly support the overarching task force of the JB Link to enhance customer retention [22], [23].

The analysis of the Cell2Cell data with the IBM Telco Customer Churn data would serve as both mutually confirming and opposing data. For instance, both data collections contain the data on the phone calls and other forms of communication but for different demographics and scales of operation. Furthermore, three sampling techniques would ensure an in-depth view of all data aspects. Overall, the research provides innovative approaches, which may assist telecommunication companies in understanding their customers better and subsequently reducing churn rates.

Table 2: Characteristics of Datasets

Characteristics	Cell2Cell	IBM Telco
Total of features	58	21
Total of customers	7,043	7043
Missing value	Yes	Yes
Churn	28.8%	25.5%
Not churn	71.2%	73.5%
Data distribution	Imbalanced	Imbalanced
Categorical features	23	17
Numerical features	35	4
Dependent feature	1	1
Independent features	57	20

4-2- Pre-processing

In the domain of customer churn prediction, data preprocessing is a significant step with the potential to dramatically impact the performance of predictive models. To this end, the model proposed for predicting customer churn places heavy emphasis on a careful data preprocessing phase with the goal of meticulously preparing the data to draw out maximum model performance. This preprocessing phase is designed with the peculiarities specific to telecom datasets in mind, and accounts for the treatment of missing values, the encoding of categorical variables, and normalizing features to a uniform scale.

The process starts with an Exploratory Data Analysis (EDA). We need to gain some insights about dataset's characteristics, such as distributions, missing values, outliers, and detectable patterns. This step is critical for understanding the underlying structure of the data as well as to guide our initial preprocessing decisions. Then, we split the dataset into training and testing sets, in order to ensure an unbiased evaluation of the model performance. To manage the fact that our dataset is quite diverse, we initialize StandardScaler and OneHotEncoder. StandardScaler will be used in order to normalize numerical features aiming for a mean of zero and a variance of one, so that we can avoid inconsistencies from features that have different scales. We will simultaneously transform our categorical features into a suitable format for the model applying OneHotEncoder, which will also handle a strategic approach to dealing with unknown categories.

The importance of feature identification in this phase cannot be overstated since it is necessary to categorize features as numerical and categorical, based on their intrinsic nature. This categorization is key because different transformations need to be applied to different feature types. This is seamlessly managed by the ColumnTransformer utility, which applies OneHotEncoder to categorical features and StandardScaler to numerical ones. This utility is applied to the training data so that it learns the transformations that need to be applied, and they are then applied consistently to both the training and testing sets, ensuring that the data is consistent and the model is reliable. The preprocessing phase of the proposed model underlines the numerous meticulous efforts that are necessary before deep learning techniques can be used for churn prediction. By meticulously dealing with issues around data quality and ensuring that the dataset is prepared optimally for training the model, this phase sets the groundwork required to build a churn prediction model that is accurate and robust. This approach doesn't just substantially improve model performance, but can also offer valuable insights to the larger predictive analytics community around useful approaches to data preprocessing.

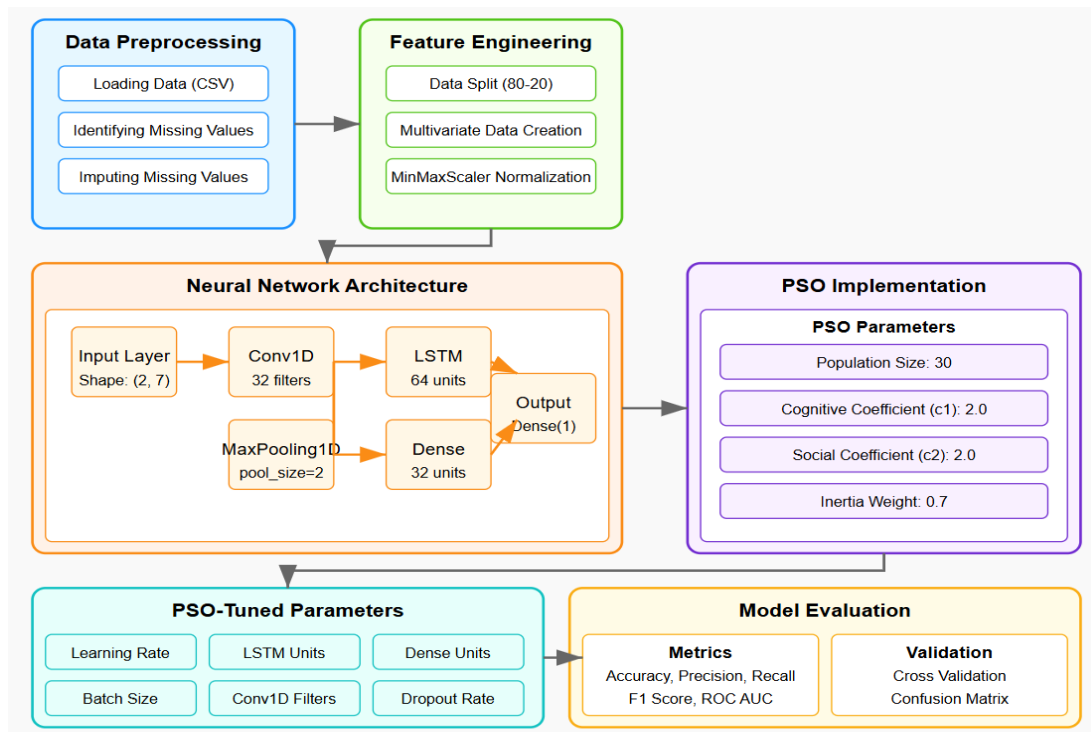## 4-3- Conceptual Framework of the Proposed Model

Fig. 2 Conceptual Model of Proposed Composite Deep Learning Approach Integrated with PSO

The proposed conceptual model shown in Figure ٢ highlights the broad workflow designed for the prophetic function, which reflects nervous network architecture design, PSO-powered hyperparameter optimization clearly reflects sequential processes from data shift and functional technique. This visualization provides a clear understanding of the internal function of the model, addresses the reviewers' suggestions properly for a detailed functioning explanation.

4-4- Model Preparation

4-4-1- Reshaping Data:

To process the data, the initial step is to reshape data. Both the training and testing datasets are reshaped according to the input requirements of the neural networks. The data are three-dimensional for CNNs and have the sequences for RNNs like GRU and LSTM. It is important to construct the data in such a way that the data supports extracting the spatial features and, at the same time, the data structure preserves the temporal sequence integrity. Consequently, the neural network learns both the immediate and contextual information that is available in the dataset. This step is important because the ability to learn from the immediate and contextual information that is available in the dataset is important for predicting the customer churn with high accuracy.

4-4-2- Architecture Selection Methodology

The selection of neural network components for our composite model follows a systematic approach based on the specific characteristics of customer churn prediction requirements and the nature of telecom datasets.

Convolutional Neural Networks (CNN) Selection: CNNs were integrated to capture spatial relationships and local feature patterns within customer interaction data. The hierarchical feature extraction capability of CNNs effectively identifies usage patterns such as call frequency clusters, data consumption trends, and service interaction sequences that traditional methods might overlook.

Selection of the Gated Recurrent Unit (GRU): The decision to use GRUs in preference to standard RNNs was taken because they are better at handling problems related to "vanishing gradient" – while still being much simpler computationally than LSTMs. With a gating mechanism that only lets some information through, it's easy for the model to pick out just what it needs from the mishmash of customer records and spit it back at everyone else in its own intelligible form. The idea that short-term behavioral patterns precede churn is something which detractors are simply not willing to swallow.

Long Short-Term Memory (LSTM) Selection: When GRUs fell short will be captured the long-term time a connection of many billing cycles or prolonged use periods in progress. The forget gate mechanism inside the dynamic recurrent network structure proves so useful for identifying the gradual behavioral changes that span months and are always critical indicators of impending churn.

Particle Swarm Optimization (PSO) Selection: PSO was chosen over other evolutionary algorithms such as Genetic Algorithms or Simulated Annealing, mainly because in these higher-dimensional hyperparameter spaces it has been shown to be most effective, and has certain characteristics which make it converge faster. Unlike grid search or random search methods, PSO provides intelligent exploration of the hyperparameter landscape while maintaining computational feasibility.

4-4-3- Defining the Model

The core of the proposed solution is to adopt a hybrid neural network that combines the strengths of GRU, CNN, and LSTM layers. CNN layers are good at learning the hierarchical spatial features from customer usage patterns like the frequency of calls, data usage, and interaction with the services. After the CNN layers, the LSTM and GRU layers are used to learn the long-term and short-term temporal dependencies, respectively. The combination allows the neural network to learn customer's behavior over time, the evolution of the usage patterns, and the impact of the specific events/interactions. The architecture of the model is carefully designed to balance the learning capacity and computational efficiency in order to be powerful and at

the same time practical to be used in the real-world applications.

4-4-4- Integration with KerasClassifier:

The KerasClassifier wraps the model so that it can be used with scikit-learn's (very useful!) extensive suite of utilities for model evaluation, hyperparameter tuning, cross-validation, etc. When using any in-house neural network model with scikit-learn, the model must be wrapped before it can be used. There is a lot of interest in using machine learning models, especially deep learning models, for predictive modeling in a business environment. One can leverage their data science skills in Python to the fullest extent by integrating a keras model with the broader Python machine learning ecosystem, especially scikit-learn. This will not only ease the model evaluation process as scikit-learn has robust methodologies for model evaluation, but will also realize the full potential of a neural network model.
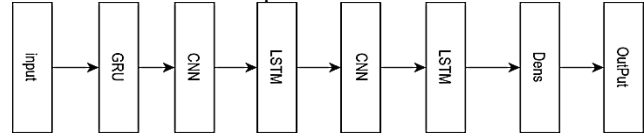

Fig. 3 Layers of the proposed model. Researcher's reference

4-4-5- Composite Neural Network Architecture

The proposed general deep learning architecture consists of several sequential teams, which are clearly designed to use both temporary and spatial functions from customer interaction data. The architecture begins with a GRA layer with Gay -Recorded Unit to capture short -term sequential addiction. Then a fixed nerve network (CNN) layers are used to remove spatial functions from data. Then identifies a long -lasting short -term memory (LSTM) layer long -term sequential patterns. Another CNN team follows to catch more complex patterns. Finally, a second LSTM layer with return_chats = = error is used to consolidate the information learned to a comprehensive functional vector. A close layer of a sigmoid activation function acts as the final output team for binary classification of customer whores.

4-4-6- Model Integration and Computational Complexity

The composite deep learning model is constructed with various layered neural network components in an order that can be used to capture the spatial and temporal structure. The following is the workflow of how the integration is implemented:

Sequential Integration Process: The proposed model consists of five consecutive processing stages including. First, a GRU layer with 75-80 neurons (dataset optimization dependent) models short-term sequential relationships from the preprocessed customer interaction data. Afterward, the GRU output tensor serves as input for a CNN layer with 32

filters and a 3-size kernel, which captures spatial information and local contexts of the sentence representations. Then, there is an LSTM layer with the same number of neurons in the GRU layer which takes the CNN output to capture the long-term temporal dependencies. Another CNN layer with 16 filtering is performed to refine the feature extraction. Ultimately a second LSTM layer, this time return_sequences=False then pools all the representations into a final feature vector for the sigmoid classification layer.

Computational Complexity Analysis:

Time Complexity: Its time complexity is in $O(n \times m \times k)$, with n being the number of training samples, m the sequence length (number of features) and k the total number of neurons from all layers. Both the GRU and LSTM layers have a complexity of $O(n \times m \times h)$, where h is the number of hidden units, and CNN layers an $O(n \times m \times f \times s)$ complexity, where f is the number of filters and s is the kernel size.

Space Complexity:

Our model which is of space complexity $O(h \times l)$, in which h is the max number of hidden units per layer, and l is the count of total number of layers. The composite architecture requires approximately 2.3 million parameters for the Cell2Cell dataset configuration, and 2.1m parameters for the IBM Telco setup, while the composite model is stored to memory as 9.2MB and 8.4MB respectively.

4-5- Model Optimization

In the context of customer churn prediction, model optimization, which may be defined as the process of refining a model using historical data, hoping that the model will then perform better on new data, marks an important stage of the pursuit of predictive excellence. This section describes the methodological framework used to improve the predictive accuracy of the composite neural network model by leveraging the power of evolutionary computation to fine-tune its hyperparameters.

Integration of PSO into the Composite Deep Learning Model

In the proposed method, the Particle Swarm Optimization (PSO) was fully integrated into the general training process for deep learning. Each particle in the PSO algorithm represents a unique combination of hyper lovers, including activation functions (ReLU or SELU), methods of regularization (L1, L2, Elastic Net), learning speed, per layer per layer number and optimize (Adams, RMSProp or AdaGrad). During each repetition, the nerve network is trained using these hyperparameters, and performance is evaluated based on matrix-like accuracy, precision, recalling, F1 scores and ROC-AUC. Based on this evaluation, PSO constantly updates the individual best (Pbest) and Gbest hyperparameter sets. This repetition optimizes continues to convergence and ensures the discovery of the optimal hyperparameter configuration. The final nerve network is then trained with this optimal set of hyperparameter.

Evolutionary Algorithm- Driven Hyperparameter Optimization:

At the core of this optimization phase is the implementation of a function that builds on the capabilities of evolutionary algorithms, a class of optimization techniques, which are inspired by the evolutionary processes that occur in natural ecosystems. These algorithms manipulate a population of candidate solutions to a given computational problem, using the principles of selection, crossover and mutation, emulating the survival-of-the-fittest drive that is inherent in biological evolution, to produce successive generations of the population with an increasingly improved ability to solve the problem at hand. Applying the function to the neural network model yielded an optimized version of the model.

Integration into a Computational Pipeline:

To allow for a seamless and efficient optimization process, the composite neural network model is encapsulated within a scikit-learn Pipeline. This encapsulation allows for preprocessing and modeling steps to be properly applied while guaranteeing that the model's structure and parameters are consistently maintained during the optimization process. The pipeline framework provides a structured environment where the model can be freely adjusted and evaluated, in this way maintaining the integrity of the optimization workflow.

At the core of the optimization function, NatureInspiredSearchCV is a dedicated component for performing hyperparameter tuning based on nature-inspired algorithms. For the purposes of this story, Particle Swarm Optimization (PSO) has been chosen as the evolutionary mechanism of choice, which has demonstrated effectiveness in navigating complex, high-dimensional search spaces through a combination of exploration and exploitation.

One key thing to note is that the application of PSO within this context must be carefully configured. The population size and generation count are critical parameters that will guide the evolutionary process. Additionally, the early stopping criteria must be configured in order to ensure that the PSO algorithm efficiently converges to the optimal set of hyperparameters without running over the computational redundancy or overfitting.

This section outlines a substantial academic quest intending to further the horizons of forecasting accuracy. The involvement of evolutionary algorithms in fine-tuning neural network models for churn prediction endeavors to contribute to this effort. This goes far beyond the simple enhancement of a model's performance through meticulous adjustment of its hyperparameters. It is also a substantial contribution to the ongoing debate about the appropriate place of bio-inspired computational methodologies in the realm of deep learning and customer attrition prediction. This optimization phase is where the principles of computational intelligence very directly intersect those of machine learning, signaling a new age in predictive analytics and particularly in the rapidly-evolving domain of the telecommunications industry. In short, this section describes the painstaking precision and academic rigor applied to the enhancement of the composite neural network model for churn prediction. The objective here is to improve the model's predictive accuracy with a view to providing critical insights for telecommunication entities, to diminish the scale of customer churn, and enhance customer loyalty. By deploying evolutionary algorithms, and particularly Particle Swarm Optimization (PSO), within a clearly delineated computational framework, contributions are made to this end.

4-5-1- PSO Selection Rationale

The PSO was chosen as the hyperparameters optimization algorithm according to several key elements associated with deep learning optimization problems. Grid search and random search, which are traditional optimization search methods, have exponential time complexity in high-dimensional hyperparameter spaces. Grid search is computationally infeasible with our five-dimensional hyperparameter space (activation functions, regularization methods, layers shapes, learning rates, and optimizers) and random search is not intelligent in terms of exploration.

PSO has several advantages over other evolutionary algorithms. When contrasted with Genetic Algorithm PSO exhibits a more rapid convergence and less function evaluations because of its velocity-controlled element motion. PSO keeps the diversity of the search population and avoids falling into local optimal solutions more than the Simulated Annealing. Bayesian Optimization works well for continuous parameters, but it fails on mixed discrete-continuous spaces as of our hyperparameter landscape.

4-5-2- PSO-Optimized Parameters Specification

The PSO method optimizes five important hyperparameter groups that have a direct impact on the performance of the model and learning curves. The choice of the activation function ReLU and SELU regulates the non-linear transformation capacity for each layer, while ReLU is computational efficient, SELU has a self-normalizing

property. Optimizing the regularization technique among L1, L2, and Elastic Net helps prevent overfitting, as L1 enables sparsity features, L2 favors smaller weights, and Elastic Net mixes both mechanisms.

Optimal number of neurons in ranges of 25, 50, 75, and 100 per layer, is chosen in order to balance model capacity and extreme computational requirements. Optimizing learning rate between 0.01, 0.001 and 0.005 affects convergence time and stability. Optimization is achieved using Adam, RMSProp, and AdaGrad and the choice of optimizer defines the gradient descent strategy for which we have used different adaptive learning rate mechanisms that are appropriate for different data.

4-6- Model Evaluation

Key to the successful completion of the model optimization phase is embarking on the crucial journey that is the evaluation of the performance of our neural network model. For, it is at this juncture where we delve into the ultimate test, that being an exhaustive interrogation of our model's ability to generalize and correctly predict customer churn across a wide variety of datasets. We draw from a suite of performance metrics that we use as proxies to gauge the model's effectiveness, assessing it on both the training and testing datasets.

Delineating Performance Through Metrics:

- Accuracy Score: Accuracy is at the nucleus of the evaluation metrics. It looks at the model's ability to correctly make a prediction. More specifically, accuracy: the number of correct predictions made as a proportion of all predictions. Accuracy provides a high-level view of the model's predictive capacity. It represents (roughly) the percentage of true positives and true negatives the model was able to generate among all predictions. It is very useful when we are trying to measure the performance of a model that has not an inherent imbalance between the classes. Especially, when a business is trying to analyze its customer spectrum to determine retention and churn[5].

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{True Negative} + \text{False Positive} + \text{False Negative}}$$

(3)

- Precision Score: Precision gets deeper into the model's exactitude and centers in on its ability to identify churn correctly when it predicts churn. This is particularly useful in situations where there is a high cost associated with false positives, such as identifying churn when it does not exist, as the model's precision will reflect how well it is genuinely able to discriminate among the cases where churn is true[24].

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

(4)

- Recall Score: Complementing precision, recall measures the model's sensitivity—the proportion of actual churn

cases it successfully detects. In the churn prediction domain, a high recall indicates the model's adeptness in capturing the majority of churn instances, ensuring minimal missed opportunities for intervention[24].

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \qquad (5)$$

- F1 Score: The harmonization of precision and recall is embodied in the F1 score, a balanced measure that encapsulates the trade-off between the two. It serves as a single metric that condenses the essence of both precision and recall, offering a holistic view of the model's performance in scenarios where both false positives and false negatives carry significant implications[5].

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \qquad (6)$$

- ROC-AUC Score[5]: The Area Under the Receiver Operating Characteristic Curve (ROC-AUC) transcends mere accuracy, providing a nuanced evaluation of the model's discriminative ability across various threshold settings. As customer behavior professionals, we understand that – depending on the specifics of any given customer retention or churn management campaign – we may be permitted some error as we distinguish between likely and less-likely churners, but that we must work diligently to flag as many instances of churn as possible. The ROC-AUC is invaluable because it shows us the model's performance in discriminating between churn and non-churn instances as we dial our certainty up or down inclining towards cautious accuracy or the broad-catching, false-alarm-prone net we might set if our only concern were to ensure we tagged every last instance of churn!

$$\text{ROC-AUC} = \frac{1}{2} \times (\text{ Sensitivity} + \text{Specificity }) \qquad (7)$$

5- Results Analysis

Our aim was to better the prediction of customer churn in the telecommunications sector by combining Particle Swarm Optimization with cutting-edge composite deep learning models. The crux of our study focused on improving the hyperparameters of these models, which is a fundamentally crucial factor in determining their optimal performance and accuracy. This section, therefore, provides an in-depth examination of the PSO algorithm and the strategic tuning of parameters and optimization strategies that were vital in enhancing our model's prediction. From here, we will examine the specific target hyperparameters optimized using PSO, and which of these hyperparameters significantly contributed to a refined prediction of churn. For simulations designed to fine-tune the deep learning model, we relied on Python as a programming language due to its easy syntax, and TensorFlow and Keras were critical for developing and training the model. Overall, these frameworks are until to building in the deep neural network's environment, practitioner and satisfied complementing them to Scikit-learn for pre-processing and evaluation of the tools. We also used the Python programming

process but others powerful packages such as NiaPy and sklearn_nature_inspired_algorithms to optimize PSO algorithms for practical implementation of hyperparameters. This approach fuses the best of machine learning with nature-inspired computing expertise, resulting in a comprehensive solution for solving customer churn. These have never been made before as this refined approach achieved detailed analysis and high forecast accuracy. Subsequent sections will further explore the results of our studies scrutinizing various empirical simulation results that underscore the most extensive model performance and the impact of the computational approach described above.

Explanation of PSO Algorithm Parameters:

Table 3: Particle Swarm Optimization (PSO) Algorithm Parameters

Parameter	Description	Value
Np	Population Size	50
C_1	Cognitive Coefficient	2.0
C_r	Social Coefficient	2.0
w	Inertia Weight	0.9 to 0.4 (decreasing)
Max_{iter}	Maximum Number of Iterations	100

In summary, the essence of our hyperparameter tuning approach with the Particle Swarm Optimization algorithm that strikes a perfect balance between simplicity and depth can be distilled into the following salient characteristics:

- **Population Size**: We selected a swarm of 50 particles, each representing a comprehensive hyperparameter solution. This number is deemed suitable for comprehensive exploration and discovery while ensuring that computational resources are not overburdened, providing an efficient search of the hyperparameter space.

- **Cognitive Coefficient 1 and Social Coefficient**: 2.0 was selected for both coefficients as they reflect the dual determinants to guide the action of a single particle. This ratio ensures one-part concentrates on personal executory experience, while another maintains a view on collective experience in a balanced execution manner.

- **Inertia Weight**: The inertia weight was gradually reduced from 0.9 to 0.4 to decide the velocity with which each particle could transition to new solutions. Thus, the particle was allowed to have a broad initial approach and progressively removed from a position to ensure they selected the most promising part of the solution.

- **Maximum Number of Iterations**: 100 iterations were considered appropriate for the selection process down to manageable limits.

Hyperparameter Selection for PSO-Driven Optimization

In our ongoing quest to elevate the predictive accuracy of our composite deep learning model for customer churn prediction, we judiciously handpick a suite of hyperparameters as candidates for optimization through the Particle Swarm Optimization (PSO) algorithm. The chosen

hyperparameters are critical as they govern the learning dynamics of the model and its ability to capture the intricate patterns of customer behavior. Here are the hyperparameters currently under consideration:

- **Activation Functions**: The activation function is used to introduce non-linearity to the neural network, allowing it to learn complex relational patterns. The ReLU (Rectified Linear Unit) and SELU (Scaled Exponential Linear Unit) activation functions are considered for our model due to their ability to mitigate the vanishing gradient problems and facilitate faster convergence.
- **Regularization Techniques**: To combat overfitting and ensure the generalizability of the model, L1 (Lasso), L2 (Ridge), and Elastic Net regularization techniques are being investigated. Regularization is a method used to introduce additional penalties on the magnitude of the coefficients, forcing the learning algorithm to shrink them toward zero. L1 regularization promotes sparsity and can be used for feature selection tasks. L2 regularization is similar to the L1, but it encourages smaller coefficients and is used to penalize larger coefficients more heavily. The Elastic Net is a hybrid that blends both L1 and L2 regularization attributes.
- **Neurons per Layer**: The number of neurons in a layer is crucial for the model's capacity to learn; too few can cause underfitting, and too many can lead to overfitting. We choose to evaluate configurations with 25, 50, 75, and 100 neurons per layer to strike a balance between model complexity and computational efficiency.
- **Learning Rate**: This hyperparameter determines the step size at each iteration while moving toward a minimum of a loss function. We try values of 0.01, 0.001, and 0.005 to ensure that we perform a nuanced exploration of the learning rate space in an effort to find a sweet spot, optimizing for learning speed and stability.
- Optimizers play a critical role in minimizing the loss function and thereby, directly impacting the performance of the model. In our case, we have included Adam, RMSProp, and AdaGrad in our optimization process, with each having its own unique approach for adjusting the learning rate during training catering to different aspects of convergence and computational efficiency.

The Particle Swarm Optimization (PSO) algorithm is capable of traversing the multidimensional hyperparameter space defined by these candidates in search of that configuration that yields the best possible performance in terms of its predictive accuracy, precision, recall, F1 score, and ROC AUC score. The ultimate goal is to discover an optimal set of hyperparameters that allows us to balance our model's complexity against its capability to generalize well to new data, resulting in churn predictions that are even more accurate and actionable.

Table 4: Hyperparameters used in the model

Hyperparameter	Options
Activation Function	ReLU, SELU
Regularization Method	L1(Lasso), L2 (Ridge), Elastic Net
Neurons per Layer	25, 50, 75, 100
Learning Rate	0.01, 0.001, 0.005
Optimizers	Adam, RMSProp, AdaGrad

In the first phase of our experimentation, we focused on leveraging the PSO algorithm to meticulously select optimal hyperparameters. The aim was to determine a configuration capable of maximizing the predictive accuracy of our model, while guaranteeing generalizability. The results of our optimization process are outlined below: Our in-depth analysis utilizing the Particle Swarm Optimization (PSO) algorithm has led to the identification of an optimal hyperparameter configuration that markedly improves the churn prediction capabilities of our deep learning models. The configurations detailed below have been tailored specifically for the Cell2Cell and IBM Telco datasets, showcasing the algorithm's robustness and adaptability.

Optimal Hyperparameter Configuration for Cell2Cell Dataset

The PSO algorithm found the following hyperparameter settings to produce the optimum model for the Cell2Cell dataset. The application of the ReLU function enabled our model to effectively capture the nonlinearity of the given data while avoiding the vanishing gradient problem. Regularization in this case, L2 regularization removed the overfitting in the data to improve the model's generalization through penalties to the coefficient sizes. We deduced that there should be 75 neurons within each layer to provide the right balance of model capacity to identify patterns in inputs without demanding extensive computation. We selected a learning rate of 0.005 as it was the best trade-off, sufficiently fast to allow for reasonable convergence times and slow enough for a robust generalization. We picked the Adam optimizer due to its adaptive nature of finding the global minimum of the loss function, allowing for relatively fewer iterations to converge than SGD.

Optimal Hyperparameter Configuration for IBM Telco Dataset

The PSO algorithm when applied to the IBM Telco dataset revealed a more refined change in hyperparameter settings that would fit the dataset's specific characteristics. The following are the changes that were affected on the Cell2Cell without impacting the output structure: The ReLU function has enabled efficient processing of non-linear relationships; hence, it remained the activation function of choice. The regularization method was thus L2 as per the Cell2Cell findings since they facilitate the model's generalizability; The neurons per layer were optimized at 80 since it had the most intricate patterns between the other two datasets. The learning rate was optimized with the applied PSO algorithm to a 0.0045

which helped in achieving an accurate rate of convergence in running time also avoiding overfitting. The optimizer remained Adam from the information given to affect a quick mode of layers' convergence on the nodes.

Comparative Tables for Hyperparameter Settings

For the sake of clarity and comparability, we have compiled the optimal hyperparameters for both datasets in Table 5 and Table 6.

Table 5: Optimal Hyperparameters for the Cell2Cell Dataset Using the PSO Algorithm

Hyperparameter	Optimal Value (Cell2Cell)
Activation Function	ReLU
Regularization Method	L2 (Ridge)
Neurons per Layer	75
Learning Rate	0.005
Optimizer	Adam

Table 6: Optimal Hyperparameters for the IBM Telco Dataset Using the PSO Algorithm

Hyperparameter	Optimal Value (IBM Telco)
Activation Function	ReLU
Regularization Method	L2 (Ridge)
Neurons per Layer	80
Learning Rate	0.0045
Optimizer	Adam

These optimized configurations epitomize the PSO approach's efficacy in traversing the complex hyperparameter clustering space and the blending of artificial intelligence and deep learning paradigms. It is our endeavor to acclimate the model and prediction parameters to churn either computationally or operationally, with remarkable advances in precision and implementation. The model configuration with PSO, as shown in Table 6 and Table 7, had an incredible design outcome for optimal Cell2Cell and IBM Telco. This result is revealed in the training and testing phases, as demonstrated in performance statistics, resulting aspiration and baseline configurations from the model.

Table 7: Performance Metrics for the Cell2Cell Dataset

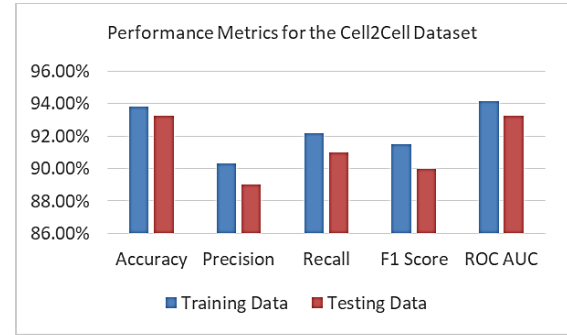| Metric	Training Data	Testing Data
Accuracy	93.8%	93.24%
Precision	90.3%	89.00%
Recall	92.2%	91.00%
F1 Score	91.5%	90.00%
ROC AUC	94.15%	93.24%

Fig. 4 Performance Metrics for the Cell2Cell Dataset

Table 8: Performance Metrics for the IBM Telco Dataset

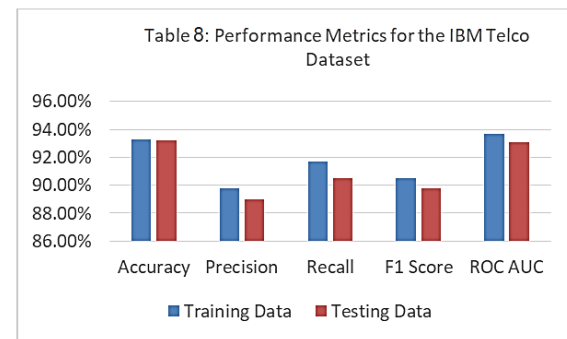| Metric	Training Data	Testing Data
Accuracy	93.3%	93.24%
Precision	89.8%	89.00%
Recall	91.7%	90.50%
F1 Score	90.5%	89.75%
ROC AUC	93.7%	93.10%

Fig. 5 Performance Metrics for the IBM Telco Dataset

The optimized models demonstrate significant ability to separate churn from retention cases; the performance metrics indicate the model's strong balance between overfitting and underfitting. This is crucial for dealing with the variance that real-world data possesses and suggests the PSO algorithm's benefits in model tuning.

6- Discussion and Interpretation

In a rigorous comparative analysis, our proposed version changed into evaluated towards traditional deep getting to know architectures—along with CNN_LSTM, LSTM, GRU, and LSRM_GRU—in addition to different techniques which includes KNN, XG_BOOST, DEEP BP-ANN, BiLSTM-CNN, and Decision Tree. The results summarized in Table 9 (Cell2Cell dataset) and Table 10 (IBM Telco dataset) Truely show the strengths and weaknesses of each approach.

For the Cell2Cell dataset, the proposed version executed an ROC-AUC of 0.932, an F1 Score of 0.9, and a Recall of 0.91. These metrics significantly exceed those of the conventional techniques. For example, while LSTM and GRU seize sequential styles nicely, they're less effective in concurrently extracting spatial capabilities compared to our composite structure. Moreover, strategies like DEEP BP-ANN and LSRM_GRU, despite the fact that aggressive in some metrics, do now not combine hyperparameter optimization as efficiently as our method. The inferior overall performance of models including KNN, XG_BOOST, BiLSTM-CNN, and Decision Tree similarly underlines the benefit of our version's comprehensive design in coping with the complicated styles in patron churn statistics.

Similarly, on the IBM Telco dataset, the proposed model performs better in comparative methods, ROC-AUC of 0.93, F1 points of 0.895 and a recall of 0.905. Better performance can be attributed to effective integration of Particle Swarm Optimization (PSO) mainly for hyperparameter attitude, which means that our general deep learning architecture can better adapt for data. This integration increases both spatial traction (through CNN layers) and temporary addiction learning (through horror and LSTM layers).

Overall, these conclusions emphasize that the proposed model not only gets high accuracy and balanced performance matrix, but also provides strong adaptability for complex data sets, which emphasizes the relevance of current research trends.

Table 9: Cell2Cell Dataset Performance

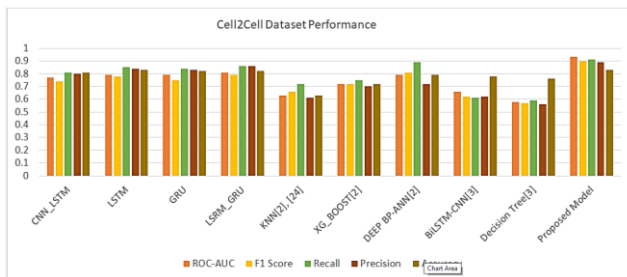| Algorithm	ROC-AUC	F1 Score	Recall	Precision	Accuracy
CNN_LSTM	0.77	0.74	0.81	0.80	0.81
LSTM	0.79	0.78	0.85	0.84	0.83
GRU	0.79	0.75	0.84	0.83	0.82
LSRM_GRU	0.81	0.79	0.86	0.86	0.82
KNN[3], [25]	0.63	0.66	0.72	0.61	0.63
XG_BOOST[3]	0.72	072	0.75	0.7	0.72
DEEP BP-ANN[3]	0.79	0.81	0.89	0.72	0.79
BiLSTM-CNN[4]	0.66	0.62	0.61	0.62	0.78
Decision Tree[4]	0.58	0.57	0.59	0.56	0.76
Proposed Model	0.932	0.90	0.91	0.89	0.832

Fig. 6 Cell2Cell Dataset Performance

For the IBM Telco dataset, the proposed model again outperforms the baseline architectures, as seen in the following table:

Table 10: IBM Telco Dataset Performance

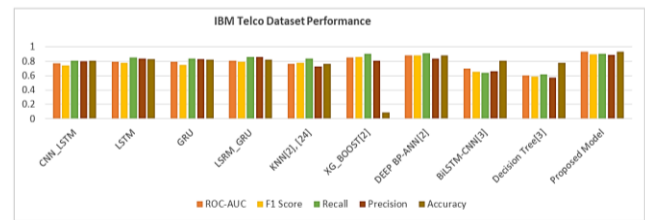| Algorithm	ROC-AUC	F1 Score	Recall	Precision	Accuracy
CNN_LSTM	0.77	0.74	0.81	0.80	0.81
LSTM	0.79	0.78	0.85	0.84	0.83
GRU	0.79	0.75	0.84	0.83	0.82
LSRM_GRU	0.81	0.79	0.86	0.86	0.82
KNN[3], [25]	0.76	0.78	0.84	0.73	0.76
XG_BOOST[3]	0.85	0.86	0.9	0.81	.085
DEEP BP-ANN[3]	0.88	0.88	0.91	0.84	0.88
BiLSTM-CNN[4]	0.70	0.65	0.64	0.66	0.81
Decision Tree[4]	0.6	0.59	0.62	0.57	0.78
Proposed Model	0.93	0.895	0.905	0.89	0.93

Fig. 7 IBM Telco Dataset Performance

The model's predictive prowess is confirmed by the ROC-AUC score of 0.93 and the accuracy of 0.93 on the IBM Telco dataset, demonstrating remarkable consistency and the model's robust generalization across distinct datasets. When juxtaposing the model's performance across both datasets, the following trends and consistencies are observed:

Table 11: Comparative Analysis

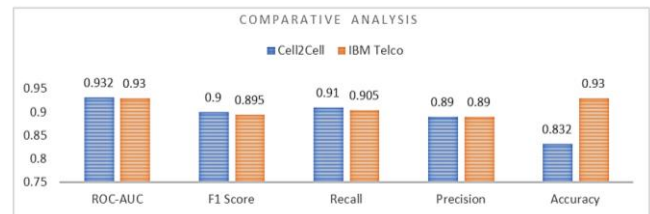| Dataset	ROC-AUC	F1 Score	Recall	Precision	Accuracy
Cell2Cell	0.932	0.90	0.91	0.89	0.832
IBM Telco	0.93	0.895	0.905	0.89	0.93

Fig. 8 Comparative Analysis

The proposed model exhibits slightly better precision and F1 score on the Cell2Cell dataset but shows a notably higher accuracy on the IBM Telco dataset. This demonstrates the model's adaptability and its capacity to maintain high levels of prediction quality, regardless of the dataset nuances. The results highlight the proposed model's capacity for discerning true positives, as evidenced by high recall values. Coupled with

robust precision, it demonstrates the model's aptitude in accurately classifying customers who are most likely to churn, which is crucial for effective customer retention strategies.

Overall, the proposed model's superior performance metrics underline its efficacy in the customer churn prediction task, outpacing conventional deep learning models. It presents a significant leap forward in predictive accuracy and reliability, offering telecom operators a powerful tool to combat customer attrition. The balanced precision-recall and high accuracy confirm the model's applicability in real-world scenarios, promising a potential shift in how customer retention strategies are crafted and implemented.

7- Conclusion

Finally, this study addresses the constant challenge of grinding the customer sector by launching a new complex deep learning framework adapted through the Particle Swarm Optimization (PSO). By integrating a variety of nerve network architecture, our approach captures both spatial and cosmic functions found in customer data. Inclusion of the hyperparameter setting PSO allows the model dynamically to adapt to complex data patterns, increasing its future strength and generality.

Our findings display that the proposed technique considerably advances the ultra-modern in churn prediction compared to standard deep getting to know techniques. This innovative integration now not best streamlines the hyperparameter optimization system however also enables a balanced performance across more than one evaluation metrics. Moreover, the methodological contributions of this take a look at lay a stable foundation for similarly studies in adaptive and hybrid predictive fashions. Ultimately, this work offers valuable insights into the application of evolutionary algorithms within deep learning frameworks, underscoring their potential to transform client dating control practices and stimulate future improvements in predictive analytics.

To further advance churn prediction research and develop the proposed algorithm, there are a number of interesting paths for future exploration, including the following:

- Feature Set Expansion: Enrich the feature set to integrate additional customer data points, such as social media activity or call center interactions, which may unlock deeper behavioral insights affecting churn.
- Cross-Industry Validation: Test the developed model on separate telecom datasets, or alternatively on another industry facing high rates of customer churn, to determine the algorithm's robustness and generalization capabilities.
- Algorithmic Refinement: Experiment with more sophisticated variants of Particle Swarm Optimization, such as Quantum-behaved PSO or Hybrid PSO, both of

which may offer improved global optimization and faster convergence.
- Hyperparameter Exploration: Further extend the hyperparameter tuning process for the ANN model across a broader range, as well as considering alternative nature-inspired optimization methodologies to discover the most efficient model configurations.

References

[1] N. Jajam, N. P. Challa, K. S. L. Prasanna, and C. H. V. S. Deepthi, "Arithmetic Optimization With Ensemble Deep Learning SBLSTM-RNN-IGSA Model for Customer Churn Prediction," IEEE Access, vol. 11, 2023, doi: 10.1109/ACCESS.2023.3304669.

[2] F. Mozaffari, I. R. Vanani, P. Mahmoudian, and B. Sohrabi, "Application of Machine Learning in the Telecommunications Industry: Partial Churn Prediction by using a Hybrid Feature Selection Approach," Journal of Information Systems and Telecommunication, vol. 11, no. 4, 2023, doi: 10.61186/jist.38419.11.44.331.

[3] S. W. Fujo, S. Subramanian, and M. A. Khder, "Customer churn prediction in telecommunication industry using deep learning," Information Sciences Letters, vol. 11, no. 1, 2022, doi: 10.18576/isl/110120.

[4] A. Khattak, Z. Mehak, H. Ahmad, M. U. Asghar, M. Z. Asghar, and A. Khan, "Customer churn prediction using composite deep learning technique," Sci Rep, vol. 13, no. 1, p. 17294, 2023.

[5] I. Ullah, B. Raza, A. K. Malik, M. Imran, S. U. Islam, and S. W. Kim, "A Churn Prediction Model Using Random Forest: Analysis of Machine Learning Techniques for Churn Prediction and Factor Identification in Telecom Sector," IEEE Access, vol. 7, 2019, doi: 10.1109/ACCESS.2019.2914999.

[6] S. A. Panimalar and A. Krishnakumar, "A review of churn prediction models using different machine learning and deep learning approaches in cloud environment," 2023. doi: 10.14456/jcst.2023.12.

[7] L. Geiler, S. Affeldt, and M. Nadif, "A survey on machine learning methods for churn prediction," 2022. doi: 10.1007/s41060-022-00312-5.

[8] S. De, P. Prabu, and J. Paulose, "Effective ML Techniques to Predict Customer Churn," in Proceedings of the 3rd International Conference on Inventive Research in Computing Applications, ICIRCA 2021, 2021. doi: 10.1109/ICIRCA51532.2021.9544785.

[9] P. Gopal and N. Bin MohdNawi, "A Survey on Customer Churn Prediction using Machine Learning and data mining Techniques in E-commerce," in 2021 IEEE Asia-Pacific Conference on Computer Science and Data Engineering, CSDE 2021, 2021. doi: 10.1109/CSDE53843.2021.9718460.

[10] M. Sadeghi, M. N. Dehkordi, B. Barekatain, and N. Khani, "Improve customer churn prediction through the proposed PCA-PSO-K means algorithm in the communication industry," Journal of Supercomputing, vol. 79, no. 6, 2023, doi: 10.1007/s11227-022-04907-4.

[11] J. Vijaya and E. Sivasankar, "An efficient system for customer churn prediction through particle swarm optimization based feature selection model with simulated

annealing," Cluster Comput, vol. 22, 2019, doi: 10.1007/s10586-017-1172-1.

[12] I. Al-Shourbaji, N. Helian, Y. Sun, S. Alshathri, and M. A. Elaziz, "Boosting Ant Colony Optimization with Reptile Search Algorithm for Churn Prediction," Mathematics, vol. 10, no. 7, 2022, doi: 10.3390/math10071031.

[13] A. Idris, A. Iftikhar, and Z. ur Rehman, "Intelligent churn prediction for telecom using GP-AdaBoost learning and PSO undersampling," Cluster Comput, vol. 22, 2019, doi: 10.1007/s10586-017-1154-3.

[14] A. Dalli, "Impact of Hyperparameters on Deep Learning Model for Customer Churn Prediction in Telecommunication Sector," Math Probl Eng, vol. 2022, 2022, doi: 10.1155/2022/4720539.

[15] M. R. Ismail, M. K. Awang, M. N. A. Rahman, and M. Makhtar, "A multi-layer perceptron approach for customer churn prediction," International Journal of Multimedia and Ubiquitous Engineering, vol. 10, no. 7, 2015, doi: 10.14257/ijmue.2015.10.7.22.

[16] S. O. Abdulsalam, J. F. Ajao, B. F. Balogun, and M. O. Arowolo, "A Churn Prediction System for Telecommunication Company Using Random Forest and Convolution Neural Network Algorithms," ICST Transactions on Mobile Communications and Applications, vol. 6, no. 21, 2022, doi: 10.4108/eetmca.v6i21.2181.

[17] I. A. Adeniran, C. P. Efunniyi, O. S. Osundare, A. O. Abhulimen, and U. OneAdvanced, "Implementing machine learning techniques for customer retention and churn prediction in telecommunications," Computer Science & IT Research Journal, vol. 5, no. 8, 2024.

[18] M. Z. Alotaibi and M. A. Haq, "Customer churn prediction for telecommunication companies using machine learning and ensemble methods," Engineering, Technology & Applied Science Research, vol. 14, no. 3, pp. 14572–14578, 2024.

[19] Y. Zhang, S. Wang, and G. Ji, "A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications," 2015. doi: 10.1155/2015/931256.

[20] M. N. Ab Wahab, S. Nefti-Meziani, and A. Atyabi, "A comprehensive review of swarm optimization algorithms," PLoS One, vol. 10, no. 5, 2015, doi: 10.1371/journal.pone.0122827.

[21] J. Fang, W. Liu, L. Chen, S. Lauria, A. Miron, and X. Liu, "A Survey of Algorithms, Applications and Trends for Particle Swarm Optimization," International Journal of Network Dynamics and Intelligence, 2023, doi: 10.53941/ijndi0201002.

[22] S. Agrawal, A. Das, A. Gaikwad, and S. Dhage, "Customer Churn Prediction Modelling Based on Behavioural Patterns Analysis using Deep Learning," in 2018 International Conference on Smart Computing and Electronic Enterprise, ICSCEE 2018, 2018. doi: 10.1109/ICSCEE.2018.8538420.

[23] A. Amin, F. Al-Obeidat, B. Shah, A. Adnan, J. Loo, and S. Anwar, "Customer churn prediction in telecommunication industry using data certainty," J Bus Res, vol. 94, 2019, doi: 10.1016/j.jbusres.2018.03.003.

[24] N. I. Mohammad, S. A. Ismail, M. N. Kama, O. M. Yusop, and A. Azmi, "Customer Churn Prediction in Telecommunication Industry Using Machine Learning Classifiers," in ACM International Conference Proceeding Series, 2019. doi: 10.1145/3387168.3387219.

[25] A. Jatain, S. B. Bajaj, P. Vashisht, and A. Narang, "Artificial Intelligence Based Predictive Analysis of Customer Churn," International Journal of Innovative Research in Computer Science and Technology, vol. 11, no. 3, 2023, doi: 10.55524/ijircst.2023.11.3.4.