ISSN: 2093-5374 / E-ISSN: 2093-5382

A Novel Approach to Transform, Store, and Analyze Real-Time Data Streams in Edge Computing Environments

Neetu Venugopal Pillai[1*], and Dr. Prashant Premji Nitnaware[2]

[1*]Research Scholar, Department of Computer Engineering, Pillai College of Engineering, Maharashtra, India. neetu.pillai012@gmail.com, https://orcid.org/0000-0002-9245-4020

[2]Professor, Department of Computer Engineering, Pillai College of Engineering, Maharashtra, India. pnitnaware@mes.ac.in, https://orcid.org/0009-0006-7573-6671

Received: September 26, 2025; Revised: November 18, 2025; Accepted: December 22, 2025; Published: March 31, 2026

Abstract

The high-speed, high-volume data used in edge computing is a fast-changing world, so the safe storage and conversion of the data to generate real-time decisions and analytics is essential. In this paper, a new solution to transforming, storing, and analyzing real-time data streams is suggested in edge computing environments. The process starts by obtaining stock data that is financial in nature followed by cleaning and normalization of the data by the Adaptive Two-Stage Unscented Kalman Filter (ATUKF). The processed data is then clustered in order to detect underlying patterns and cluster like points. The Finite Basis Physics-Informed Neural Networks (FBPINN) approach is used to optimize service interactions and minimise service response time and delays during data transmission in order to reduce latency. The method is executed in Python and experiments prove it to have a 99.2 % accuracy with a computation time of 1.150 seconds. The suggested approach is much better than the current approaches, such as Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), and Deep Neural Networks (DNN). This method can be used in high-frequency trading and other applications that need real-time finances data processing, as it is a more efficient and scalable solution that can be achieved by utilizing edge computing and sophisticated machine learning methods. The findings demonstrate the capability of combining ATUKF to process the data, clustering to identify the patterns, and FBPINN to optimize the response of the service as an overall structure to improve the performance of the edge-based real-time data analytics.

Keywords: Edge Computing, Adaptive Two-Stage Unscented Kalman Filter, Finite Basis Physics-Informed Neural Networks.

1 Introduction

The era of electronic transformation has brought about an unprecedented surge in the volume, velocity, and variety of data generated across diverse domains (Kumar et al., 2021). This explosion of data is further fueled by the widespread deployment of edge devices and sensors operating at the network periphery (Li et al., 2021). Constant data streams such as financial stock data have become essential inputs into real-time decision-making in areas like financial markets, financial stock data processing, smart infrastructure, and industrial automation (Bandi, 2021; Abdellatif et al., 2021). The challenge of

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), volume: 17, number: 1 (March-2026), pp. 191-208. DOI: *10.58346/JOWUA.2026.I1.012*

*Corresponding author: Research Scholar, Department of Computer Engineering, Pillai College of Engineering, Maharashtra, India.

having to process financial stock data quickly, with the demand of an immediate response, has brought to the fore the shortcomings of more conventional centralized computing models used in the solution of latency-sensitive problems. The importance of edge computing lies in the fact that it allows for decentralizing computing and storage, which is found at the sites of financial stocks (Dautov et al., 2021). This improves the transmission delays, and faster decision-making is made (Luo, 2023). The closeness enhances the responsiveness of systems and new opportunities towards timely analytics during high-frequency financial stock data processing (Aldhyani & Alzahrani, 2022). Both proper transformation and storage of financial stocks at the edge enhance efficiency in processing new information as well as enable organizations to make sound decisions in dynamic and high-pressure trading. Therefore, intelligent, scalable, and responsive edge computing solutions are gaining significant importance in the quest by industries to remain competitive and agile (Yemunarane et al., 2024; Moseley et al., 2023).

However, edge computing still has significant limitations when dealing with large volumes of stock market data that is of high speed (Gülmez, 2023). The traditional methods can prove to be ineffective because of storage inefficiencies, high calculation costs, and the translation of complex or multidimensional data. Moreover, clustering models used in edge environments fail to calculate the correct parameters to use in optimum stock data grouping of financial data (Egrioglu et al., 2022). The proposed study aims to close these gaps by creating a robust framework that will improve financial stock data storage management, facilitate its transformation, and support real-time analytics with better efficiency (Gunalan et al., 2022). Overcoming these essential issues, the proposed work will be able to advance the limits of what can be achieved within the context of the edge-based financial stock data processing.

Key Contributions

1. To allow real-time processing of financial stock data on an edge computer, the paper integrates both edge computing and deep learning models in order to decrease the latency and enhance prediction efficiency in high-frequency trading.
2. It presents the application of the Adaptive Two-Stage Unscented Kalman Filter (ATUKF) to preprocess the data and achieve the highest efficiency in the interaction of data and services with the help of FinBPINN (Finite Basis Physics-Informed Neural Networks).
3. The study deals with the problem of high-bandwidth and low-latency through the usage of wireless communication and edge devices that allow for faster decisions and enhance the scalability of financial prediction systems.

The paper is structured in the following way: Section 1 provides the introduction to the relevance of edge computing to the processing of financial stock data. In Section 2, reviewed methods in the literature will be utilized to detect gaps in the analysis of real-time stock data. Section 3 introduces the researcher's methodology for gathering and pre-processing financial stock data and optimizing it. Section 4 provides the findings of the suggested approach. Lastly, the findings are addressed in Section 5, which points to the network bandwidth and latency trade-offs, which play an essential role in wireless and ubiquitous computing. At the end, Section 6 sums up the paper and proposes the findings and future enhancements of financial stock data processing.

2 Literature Review

The authors Akbar et al., (2024) suggested a safe deduplication mechanism of storage to enhance efficiency by cutting down the bandwidth and the storage charges. It is a hybrid algorithm that uses the Two Threshold Two Divisor (TTTD) algorithm with Dynamic Prime Coding (DPC) to minimize deduplication information, and it minimizes data privacy. It boosts the throughput and processing speed and reduces the computational cost of the hash function to 772 ms, which is better than the current methods.

Transformer deep learning frameworks were employed (Wang et al., 2022) to predict stock data indices related to finances. The multi-head attention mechanism of Transformer and encoder-decoder design were initially introduced to process natural language, but were used to enhance the precision of time series prediction regarding the financial data. The paper concentrated on the prediction of one financial stock index, yet it recognized the high levels of interdependence among the global financial markets. The novelty of the work is that it incorporates the real-time method of Intrusion Detection System (IDS). Also, similar terms to neural networks, Deep Learning (DL), and Bayesian networks were investigated (Sankaranarayanan et al., 2020).

In the paper of (Gao et al., 2022), a deep learning (DL) method along with genetic algorithms was suggested as the predictor of the direction of the overnight returns to a particular financial stock index, where global stock market indexes are used as the sources of data. Although the approach has potential, it is limited. The paper admits that economic status, nature of industries, investor attitudes, and political forces can also affect the movement of stock market index (SMI) and proposes that in future studies such factors should be included in the model. Also, it is intended to enhance the computational efficiency of the proposed methods.

Wang & Wang, (2022) have examined blockchain technology and supply chain financing. In addition to the unique circumstances surrounding supply chain finance at this time, an analysis was done on the risk control system, supply chain's cash flow, and management system. All stakeholders optimize the supply chain finance risk control system while reducing costs and increasing company efficiency in order to drastically reduce the risks encountered by each party.

Zeng, (2022) built an Internet of Things (IoT) solution to enhance financial management and early detection of financial risks, as well as increase the risk resistance of businesses. The financial risk early-warning models that were founded on accounting data were analyzed, and the Backpropagation neural networks (BPNN) mined the financial stock data.

The integration of mobile edge computing (MEC) services aimed to enhance data processing speed and accuracy. An edge service preloading optimization model was also created using Geographic Point of Interest Information and BPNN.

Park et al., (2022) proposed the LSTM Forest system, which involves a Random Forest (RF) and Long Short-Term Memory (LSTM) when forecasting financial stock returns, and a multi-task model to classify the direction of returns and predict stock returns. In a similar manner, Ansari et al., (2024) defined a Reinforcement Learning agent with Advantage Actor-Critic (A2C) and Deep Deterministic Policy Gradient (DDPG) algorithms to trade in multiple stocks, which were confirmed with 30 Dow Jones equities and performed better than the benchmark Dow Jones Industrial Average index during backtesting. The main issue in the strategy was the speed of execution and accuracy to promote effective financial data processing and trading strategies.

The literature indicates that there is a significant development in the field of financial stock data processing on the basis of deep learning and edge computing methodologies (Udayakumar et al., 2023).

Such methods as data deduplication and Transformer-based model can enhance data processing, precision of predictions, and genetic algorithm can be used to optimize it further. The idea of combining blockchain technology and the Internet of Things (IoT) is expected to improve the speed of financial risk management and data processing. LSTM-Random Forest and Reinforcement versions of the hybrid models maximize stock forecasts and market trades. These papers explain why better computational efficiency, accuracy of data, and scalability are needed, which is consistent with the research objective of incorporating edge computing to achieve faster, more efficient predictions.

2.1 Research Gap

Current deep learning financial stock data in edge computing has drawbacks, such as high computational expenses, reduced generalizability, and inadequate real-time adaptability. Such models as LSTM and Transformer do not take into account macroeconomic factors or interdependence across markets. Deduplication techniques have privacy problems, whereas risk models are not that robust on varied data. Trading systems that are based on reinforcement learning also have difficulties in terms of competing speed and precision. Such challenges form the basis of a combined, low-latency, explainable deep learning system capable of scaling, being adaptable, and contextually aware in the edge environment to process financial stock data.

3 Proposed Methodology

The proposed method is depicted in figure 1 and involves the following steps: firstly, the financial stock data are collected, and secondly, the ATUKF is used to clean the data and normalize it. Where, a model scrutinizes and groups data, depicts the deliberate intrusion of debatable or misleading information into the visual data and protect the integrity of data identification systems. These machines are used in cleaning and preprocessing data, clustering, and real-time analytics. The processed data is then clustered to give patterns, and the FBPINN method is used to reduce the time of response of service as well as the delay in transmission of data. It uses wireless communication technologies in order to provide low-latency and high-bandwidth data transfer and hence is applicable in real-time financial stock data processing.

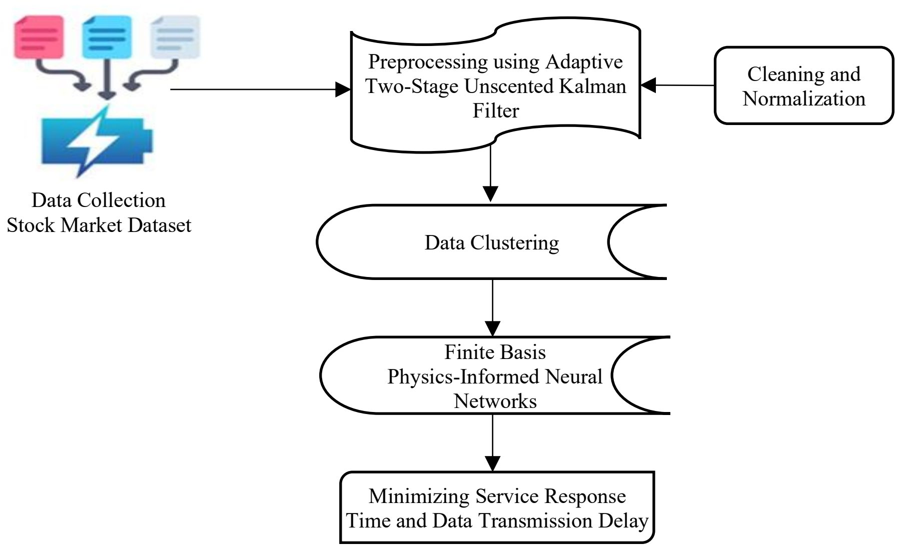

Figure 1: Block diagram of the proposed method

In order to meet the aims of the design, analysis and modeling of the storage and transformation techniques of high volume and high velocity financial stock data streams in the stock exchange field, a comprehensive approach is put forward whereby the start point is the collection and pre-processing of real-time financial stock data through APIs to the end point where trends of financial stock data are predicted. Also, the methodology involves the optimization of wireless communication in the efficient delivery of the financial stock data to provide the real-time market data in a way that it can be analyzed and used to make a decision. Real-time processing of data is essential, particularly in financial stock data, where one is required to make decisions within a short period of time in order to take advantage of the market trends.

3.1 Storage Analysis Technique

Within the framework of edge computing, the storage analysis is an integral part of the management of the financial stock data across distributed and resource-constrained nodes. The suggested model has a multi-layered framework of storage analysis that aims at maximizing financial stock data management, increasing the resilience of the system, and increasing its computational efficiency on the edge. The framework starts with the tiering of data, whereby financial stock data are differentiated into separate groups in terms of frequency of access. The data that is accessed the most (hot) in finance is stored in high-speed, low-latency storage modules on the edge nodes, and the data that is accessed the least (warm and cold) is gradually offloaded to secondary storage or remote cloud layers. This stratification decreases the delays caused by access and also guarantees optimal use of the limited resources of local storage.

The model is also augmented with predictive storage analytics to help support operational stability through the application of machine learning-based forecasting to predict failures in data storage of financial stock, saturation points, and bandwidth thresholds. These analytics assess the existing patterns of consistency in financial stock data storage history and predict further conditions of storage health, thus permitting proactive data transfer, dynamic buffer placement, and responsive storage load distribution. The system also uses storage usage profiling, which constantly tracks storage parameters, read/write intensity, capacity limits, frequency access maps, and wear indicators unique to a particular device (particularly NAND-based flash memory). This profiling helps to determine the stress areas of storage and proactively manipulate the data routing functionalities of the edge network of financial stock data.

Besides, the model uses data deduplication and a compression algorithm to reduce redundancy and optimize the memory usage of financial stock data. These methods use a block-level comparison of data and also apply entropy-based compression to compress the data based on the unique data fragments and decrease the total data size. At the same time, the content-aware storage policies are used to group the incoming financial stock data streams by metadata attributes (e.g., format, origin, sensitivity), and allow the application of differentiated retention, encryption, and access control policies. Lastly, it is complemented with the edge-to-cloud synchronization consistency analysis that guarantees that the stock information that is replicated into the heterogeneous storage settings does not lose its structural and semantic integrity. All these techniques of storage analysis form a smart, autonomous storage management layer that is specific to the needs of financial stock data processing on the edge computing environment.

3.2 Data Collection

The financial stock market data consists of historical daily price data of all tickers on NASDAQ (www.kaggle.com - stock-market-dataset), which were retrieved with the finance Python package until

April 1, 2020. Stocks or ETFs have a CSV file in which each row has a field of Volume, Adjusted Close, Low, High, Close, Open, and Date, and the name of the file is a ticker symbol. Other metadata, such as company names, can be found in the symbols_valid_meta.csv file. The data, which was collected via APIs in big stock exchanges, offers high-frequency and minute-based data gathering with parameters such as Volume, VWAP, Open, Close, High, Low, Number of Trades, and Timestamps. The wireless communication provides an efficient way of transmission of the data to the edge nodes with minimum delay, and the delay of data collection is recorded to test the performance in order to perform the financial analysis in detail.

3.3. Pre-Processing Using Adaptive Two-Stage Unscented Kalman Filter

In this sector, the pre-processing of the data is done through ATUKF (Hou et al., 2024). The ATUKF is employed in cleaning and normalization of the financial stock data. An Adaptive Two-Stage Unscented Kalman Filter (ATUKF) is adopted to clean and normalise financial stocks information. This filter is beneficial when dealing with non-linear, noisy details, as in the case of stock market data streams. The ATUKF processing steps can be separated into two significant steps: prediction and update, in which there is an iterative correction of the estimates by the algorithm using noisy observations. The equations of the ATUKF model that are employed in this process are as follows:

Prediction Step

$$\hat{x}_k = A\hat{x}_{k-1} + Bu_k \tag{1}$$

As shown in equation (1), the prediction step of the Kalman filter is where x h k is the predicted state at time k, using the past state x h (k-1). The state transition is represented by the matrix A, and the prediction is modified with the help of the control input u k. This step predicts the state of the system, which is essential in the filtration of noise data in financial stock data analysis.

$$P_k = AP_{k-1}A^T + Q \tag{2}$$

The error covariance predicted at time k is computed in equation (2). It corrects the uncertainty of the expected state of the forecast using the past error covariance P (k-1) and process noise Q. This is in order to determine how reliable the forecast is in such systems as financial stock information processing.

Update Step

$$K_k = P_k H^T (HP_k H^T + R)^{-1} \tag{3}$$

In equation (3), the Kalman gain K k is computed and it is the weight to be assigned on the new measurement in the update step. It involves an error covariance prognostication Pk, the prediction model H, and the measurement noise covariance R. This gain is the best fit of the predicted state in accordance with the difference between the actual and the expected state of the financial stock, enhancing better prediction of the financial stock data.

$$\hat{x}_k = \hat{x}_k + K_k(z_k - H\hat{x}_k) \tag{4}$$

In the Kalman filter, the correction of the predicted state x wide hat at a particular step k was presented in equation (4), in which the Kalman gain K k was used to correct the state. In this case, z k is the new measurement, and Hx H x k is the forecasted measurement. K k is used to weight the difference between the observed and projected measurements to update the state estimate to enhance the quality of financial stock data predictions.

$$P_k = (I - K_k H)P_k \tag{5}$$

The Kalman filter is an update process that updates the error covariance P k after the update step which is given by equation (5). It makes corrections on the predicted error covariance by minimizing it under the Kalman gain K k and the measurement model H. The equation guarantees that the uncertainty of the state estimate is minimized upon the addition of the new measurement, which is critical in the accurate analysis of financial stock data. The equations are applied to the edge devices to remove noise and enhance the financial stock data for clustering and analysis by improving the accuracy of the data in the process. The active flexibility of ATUKF is adapted to the edge-setting environments in which data may be unpredictable and noisy.

Compared to traditional approaches, ATUKF has a two-step process that deals with uncertainties of processes and measurements separately, which makes more accurate estimations. Its adaptive character enables it to change its parameters with the dynamism of the data quality that is well-suited in edge cases where data changing variability and limited computing resources are the general rule. The methodology guarantees improved reliability and accuracy in preprocessing financial stocks data, which forms a strong basis of downstream analytics and decision-making in edge-based complex and real-time scenarios characterized by high latency and bandwidth thresholds, which can be represented in equation (6).

$$x_k = d(y_k, u_k) + Ca_k + v_k k \geq \tau \tag{6}$$

Where, $x_k = [x_{1,k}, x_{2,k}, \ldots . x_{N,k}]^T$represents the measurement vector, $d(y_k, u_k)$ is the measurement function, C is the Visualize the areas of the data model focused on for its choice, a_k represents robustness against generated synthetic data at time$k$, $\tau$is the scaling parameters. The ATUKF successfully removes noise in the raw data and, therefore, gives more precise results and reduces the possibility of false negatives or false positives in intrusion detection, as shown in equation (7).

$$\begin{cases} y_{k+1} = f(y_k, u_k) + Ga_k + g_k \\ x_k = d(y_k, u_k) + Ca_k + v_k \end{cases} \tag{7}$$

Where represents a matrix which describes the distribution of adversarial perturbations directed at deceiving data, is the process noise at time, is the data acquisition function, is the system functional, and is the discrete nature of the measured data. ATUKF makes sure the data is cleaned by use of detection and correction of any irregularities and inconsistencies to ensure that only quality data is utilized in subsequent analysis and decision making, as indicated in equation (8)

$$\hat{y}_{k+1|k+1} = \hat{y}_{k+1|k+1} + \beta_{k+1|k+1}\hat{b}_{k+1|k+1} \tag{8}$$

Where is a model that analyses and categorizes data, represents the intentional insertion of controversial or misleading data into the visual data, and safeguards the integrity of data identification systems. Normalizing the input data, ATUKF ensures all features are on a consistent scale, which improves the performance and speed of models used in intrusion detection and prevention, as represented in equation (9)

$$w_k = \sum_{j=0}^{2n} M_j^s f(x_k^j) \tag{9}$$

Wherew_k the leads measurement matrixM_j^s is the process-generated synthetic data, the adaptive nature of ATUKF allows it to adjust dynamically to changes in the data, making it highly suitable for industrial environments where sensor data might fluctuate due to varying operational conditions. Finally, ATUKF has successfully cleaned data and normalized the input data. Afterward, ATUKF pre-processed data is fed into the data clustering segment.

3.4 Data Clustering

The normalized financial stock data is clustered with the help of the K-Means clustering algorithm to show the pattern and cluster similar data points, and make the analysis more interpretable. The information is grouped into designated sets, and the findings are stored as a CSV file. Scatter plots are created in order to illustrate the points and cluster centers. Determination of the optimum number of clusters is done through Elbow Method by analyzing the inertia values. Also, computational costs and response times are estimated and plotted, showing the efficiency of managing the data in high velocity streams, especially in an edge computing environment where there is wireless communication to update the information in real-time.

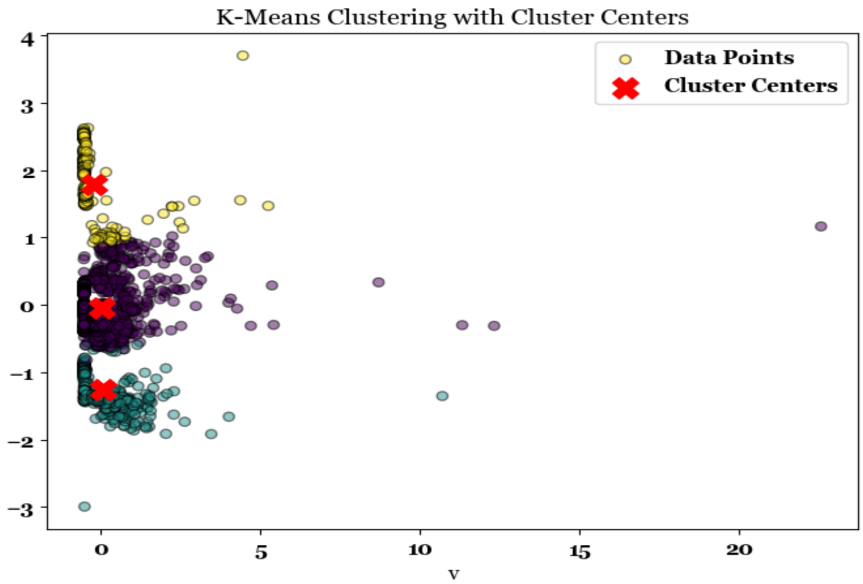

Figure 2: K-Means clustering with cluster centers

The findings of the K-Means clustering of financial stocks data, i.e., the v and w features, are visualized in figure 2, and three clusters are formed. The majority of the clusters are also clustered in the bottom-left part of the plot, with a few points being spread in the further areas. There are those clusters that are more spread and those that are smaller. Some of the points are far away from the centre of the cluster, and this may be an indication of outliers or misfit data. The key points are identified with the help of the red "X" marks to be further analyzed. Such a clustering technique is helpful in learning about the market behavior and finding patterns to trade with the financial stock information.

3.5. Interacting Services Deployment Using Finite Basis Physics-Informed Neural Networks

This part addresses the application of FBPINN to implement the interacting services by minimizing service response times on one hand and the monetary stock data transmission delay amongst services on the other hand. Optimization of financial stock data transmission and service interactions is done on the edge network using the FBPINN (Finite Basis Physics-Informed Neural Network). The FBPINN model integrates the predictive nature of neural networks and physical laws so as to minimize delays in data transmission in the system, as well as maximize the responsiveness of the system. The principal processing in FBPINN is the incorporation of the physical constraints in the training process of the neural network, so that the predictions of the system are in line with the real behaviours. The FBPINN architecture can be mathematically modelled as follows:

$$\mathcal{L}_{\text{FBPINN}} = \sum_i (\| \hat{y}_i - y_i \|^2 + \lambda \| \mathcal{P}(\hat{y}_i) - \hat{y}_i \|^2) \tag{10}$$

In equation (10) y ̂i represents the output of the neural network as predicted, y is the actual data (ground truth), $\mathcal{P}$ (y ̂i) is the physical law that the expected output is supposed to satisfy (e.g., conservation laws, physical constraints), λ is a regularization parameter, balancing the loss because of prediction and the loss because of breach of physical laws.

This equation makes sure that the output of the neural network is not only accurate but also follows the physical laws of stock data patterns of financial data, which enhances the quality of the financial stock data predictions and makes the system reliable. The FBPINN eliminates the delays of data transmission throughout the network by maximizing the response time of services.

The FBPINNs integrate the neural networks and laws of physics to reduce the delays of the data transmission and the response time of the services, which are essential in the processing of the financial stock data in real time. The strategy will incorporate the predictive capabilities and flexibility of neural networks and the accuracy of physics-based models to increase the efficacy and performance of the system. FBPINNs will enable timely decision-making and a more regular provision of services since the delays will be reduced. The neural network structure integrates the law of physics or system constraints, which leaves it efficient in an edge environment that may demand low latency and high bandwidth, as illustrated in equation (11).

$$\overline{NN}(g;\theta) = \sum_i^p \omega_i\,(g) \cdot unnorm \circ NN_i \circ norm_i(g) \tag{11}$$

Here,$NN(X;\theta)$ denotes a distinct neural network positioned on every subdomain,$\omega_i(g)$ denotes the locally confining window function, smooth and differentiable, for every network within its subdomain,nom_i denotes that the input vector$g$ has been separately normalized in each subdomain,$unnorm$ denotes the standard normalization that is applied to every output of a neural network, and the value of$\theta = \{\theta_i\}$. The Finite Basis approach is integrated to ensure that the model aligns with physical behaviour or principles, as given in equation (12).

$$\omega_i(g) = \prod_j^e \varphi\left((g^j - x_i^j)/\sigma_i^j\right)\varphi\left((y_i^j - g^j)/\sigma_i^j\right) \tag{12}$$

Here, the midpoints of each dimension's overlapping sections on the left and right are represented by x_i^jandy_i^j;j they stand for each input vector dimension. Conversely, σ_i^jdenotes the collection of parameters that are specified beyond the overlap zone. The model accounts for both spatial and temporal correlations in vehicle and cloud behaviours, as given in equation (13).

$$L(\theta) = L_S(\theta) \tag{13}$$

Here$L$, it is denoted as one of the environmental factors and Srepresents the coefficient for each basis function. Finite basis functions are integrated into the model to ensure that the neural network's predictions are consistent with the underlying physical phenomena, as given in equation (14).

$$u(g) = \omega_1\, cos(\omega_1 g) + \omega_2\, cos(\omega_2 g) \tag{14}$$

Here, $u$denotes multi-scale frequency components to choose ω_1 ω_2, that is, there are both high- and low-frequency components in the solution. The trained FBPINN is used to optimize service interactions by adjusting parameters and configurations to minimize service response time and data transmission delays in the following equation (15)

$$u(g_1, g_2) = \frac{1}{\omega} sin(\omega g_1) + \frac{1}{\omega} sin(\omega g_2) \tag{15}$$

Here$g_1 and g_2$ is the physics-informed parameter that penalizes deviations from physical laws, $\frac{1}{\omega}$Which denotes the service interaction parameter. Finally, the FBPINN deployed the interacting services, thereby minimizing service response time and data transmission delays between services.

Algorithm 1: Real-Time Financial Stock Data Processing in Edge Computing

Input:

D = {X, Y}: Training dataset with inputs Xand labels Y

ATUKF: Adaptive Two-Stage Unscented Kalman Filter for data cleaning and normalization

Φ: Set of logical constraints for system behavior and network performance

FBPINN: Finite Basis Physics-Informed Neural Network for interacting service deployment

η: Learning rate for neural network optimization

T: Number of epochs for training the model

λ: Regularization factor for minimizing physical law violations

Output:

Trained and optimized model θ for predicting financial stock trends with minimal latency.

Pseudocode:

1. Initialize model parameters θ, including weights and biases for the neural network
2. Initialize the Adaptive Two-Stage Unscented Kalman Filter (ATUKF) for data preprocessing
3. Initialize Finite Basis Physics-Informed Neural Network (FBPINN) for service deployment
4. For each epoch t = 1 to T:
  a. For each data point (x, y) in D:
    i. Clean and normalize data using ATUKF
    ii. Apply data clustering to identify patterns using K-Means
    iii. Pass the input x through the model to predict stock prices
    iv. If prediction is successful:
      - Process data and update model predictions
    v. Else:
      - Log error and reattempt prediction
  b. Optimize data transmission and minimize service response time using FBPINN:
    i. Integrate physical constraints into the neural network to optimize the system's behavior
    ii. Update the model based on physical law adherence
  c. Calculate task loss:
    i. Prediction loss: Lprediction = MeanSquaredError(f(x), y)
    *ii. Regularization loss: Lregularization = λ * Violation(f(x), θ)*
  d. Compute the total loss:
    Ltotal = Lprediction + Lregularization
  e. Update model parameters θ via backpropagation:
    *θ ← θ - η * ∇θ Ltotal*

5. Return the trained model θ with optimized predictions and minimal latency

In the edge computing algorithm 1 processing under real-time financial stocks data, firstly, the model parameters, neural network weights, and the Adaptive Two-Stage Unscented Kalman Filter (ATUKF) are initially initialized. It then feeds the input data in multiple epochs, where each point of data is purged and normalized by ATUKF, grouped, and fed to the model to be forecasted. The predictions of the model are optimized with the help of Finite Basis Physics-Informed Neural Networks (FBPINN) in order to reach the service response times and the real-world behavior. The task loss, comprising prediction loss as well as regularization loss, is calculated and utilized to update the model parameters through backpropagation. The process is iterative, which guarantees that the model becomes trained to predict financial stock trends as accurately as possible and incur minimal latency, as well as improve data transmission efficiency in the edge environments.

4 Results

The suggested methodology is realized with the use of Python and deep learning models (LMST, Transformer) with the help of TensorFlow and Keras, machine learning algorithms (Random Forest, genetic algorithms), and data manipulation with the help of Pandas. The visualization is performed with the help of Matplotlib and Seaborn, and NumPy is used to do numerical calculations. Scalability is done using cloud services such as AWS or Google Cloud, and model experimentation and testing are done using Jupyter Notebook.

The use of key parameters in the experiments was to optimize the performance of the model. The learning rate was 0.001, 50 epochs, and a batch size of 32. Adam optimizer was employed and dropout rate was set to 0.2 to eliminate overfitting. The LSTM model consisted of 3 hidden layers of 128 units, and the model with Random Forest had 100 trees. The process and measurement noise levels were configured to 0.1 in the Kalman Filter and 20 batch of normalization and ReLU activation were used. The K-Means clustering was launched using 5 clusters to cluster financial data.

4.1. Performance Measures

It is a significant step in the calculation of the most perfect forecast. Measures of performance, such as ROC, precision, accuracy, and recall, are measured to assess performance.

4.1.1. Accuracy

Accuracy of a measurement reflects the similarity of the measurement to the actual value, that is, there is only small error or variance. It has been expressed as given in equation (16),

$$Accuracy = \frac{TP+TN}{TP+TN+FN+FP} \tag{16}$$

In equation (16), the accuracy is determined by the ratio of the correct predictions (True Positives (TP)) and True Negatives (TN) to the total number of predictions (True Positives (TP) + True Negatives (TN) + False Negatives (FN) + False Positives (FP)). It measures the general performance of a classification model, which is vital in determining the financial stock data prediction models.

4.1.2. Precision

Precision is calculated by dividing the number of true positives by the number of false positives, and it is given by the equation (17),

$$precision = \frac{TN}{FP+TN} \tag{17}$$

Precision is computed using equation (17) and is the ratio of True Positives to the summation of True Positives and the total number of the False Positives. It determines the accuracy of positive predictions, and this means the number of predicted positives that were accurate. The measure is significant in measuring the performance of financial stock data analysis models.

4.1.3. Recall

The recall measures the ability of a machine learning model to identify good instances. That is, it determines the likelihood of getting a favourable result. That's provided in equation (18)

$$Re\,c\,all = \frac{TP}{(TP+FN)} \tag{18}$$

The recall can be calculated using equation (18) as the ratio of True Positives (TP) to the number of True Positives (TP) + False Negatives (FN). It quantifies the degree to which the model identifies actual positive instances that are real, therefore showing how well it represents all the important data points. The measure is critical in the financial analysis of stocks where one would want to minimize on missed opportunities (false negatives).

4.1.4. ROC Curve

It is the ratio of area of false negatives to the area of true positives, and it is represented by equation (19).

$$ROC = 0.5 \times \left(\frac{TP}{TP+FP} + \frac{TN}{TN+FP}\right) \tag{19}$$

TPR (sensitivity) and TNR (specificity) are each applied with the other to estimate how a model correctly finds a positive or a negative case. This composite measure is significant in stock analysis on financial data because it is inevitable to reduce the false positives and false negatives in order to make better decisions.

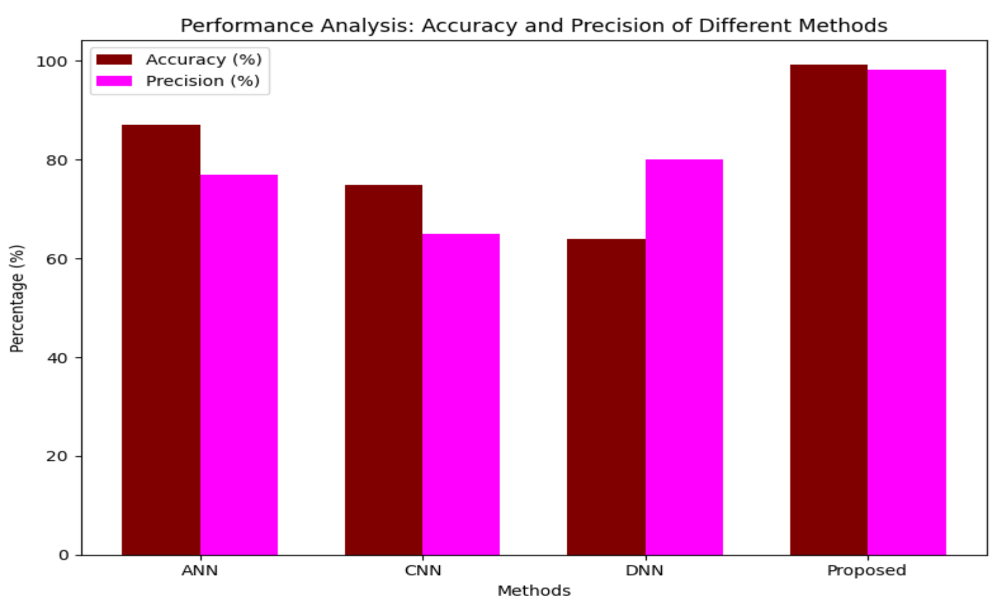

Figure 3: Performance comparison of accuracy and precision for different methods

Figure 3 shows the Proposed Method achieves the highest performance with an accuracy of 99.2% and precision of 98.2%. ANN method has a 87 percent accuracy and 77 percent precision. The accuracy and precision of the DNN method are 64 and 80 respectively whereas the accuracy and precision of CNN is 75 and 65 respectively. In both metrics, the Proposed Method has been found to be superior amongst all other methods.

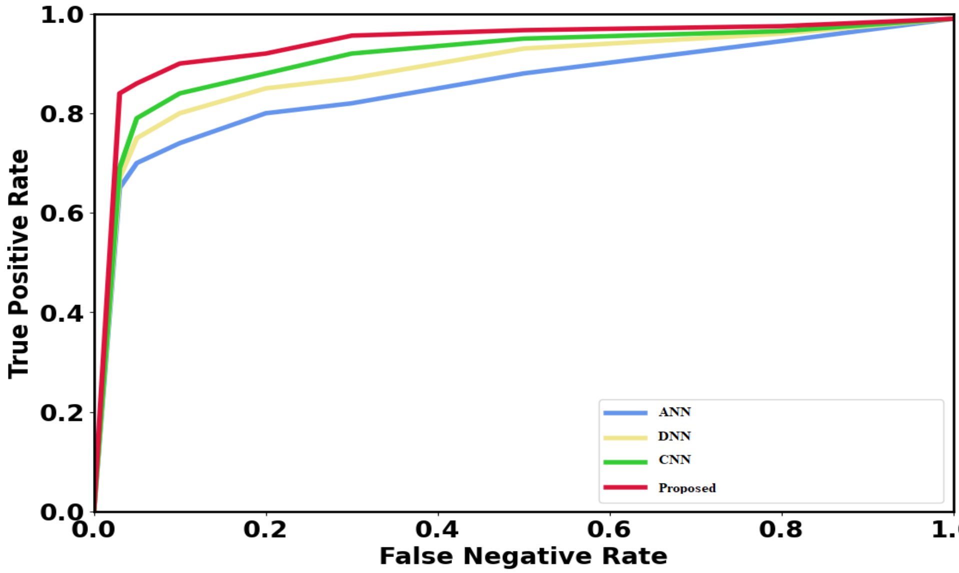

Figure 4: Analysis of the performance of the ROC value with proposed and existing methods

Figure 4 represents ROC analysis of the proposed method in the comparison with ANN, DNN, and CNN. The highest actual positive rate is obtained at all the false negative rates with the proposed method, which implies better classification performance. It also has a higher detection accuracy and robustness compared to current practices, where CNN and DNN demonstrate moderate performance, and ANN depicts the worst performance.

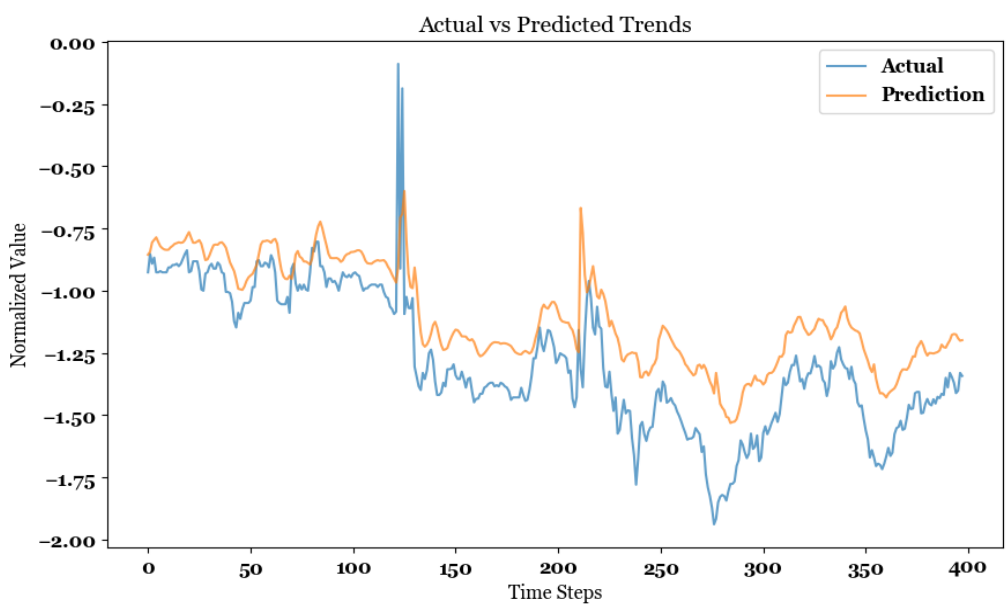

Figure 5: Actual versus predicted trends

Figure 5 presents a comparison of the predictions with the actual data of the model, and it is observed that the predicted values (orange line) mostly follow the actual values (blue line), which means that the model is working well in the overall tendency. But there are exceptions, e.g., at time steps, say, 150, there is an overshoot in the model. Although the model is helpful in tracing broad trends, it might not be able to track abrupt rises and falls.

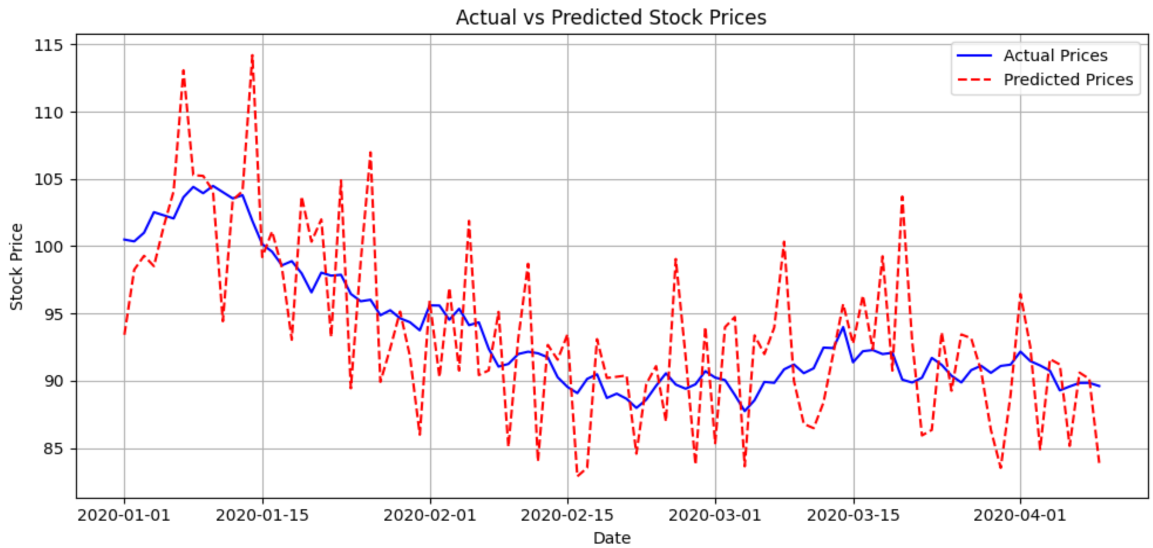

Figure 6: Actual vs predicted stock prices

Figure 6 shows the comparison between the actual stock prices (blue line) and the prediction of stock prices (red dashed line) over time. The graph is used to evaluate the accuracy of the model as it indicates the similarity between the fundamental changes in the stock market and the predictions suggested by the model. It also points out the differences between the expected and the real values, which means that there are areas in which the model can be improved.

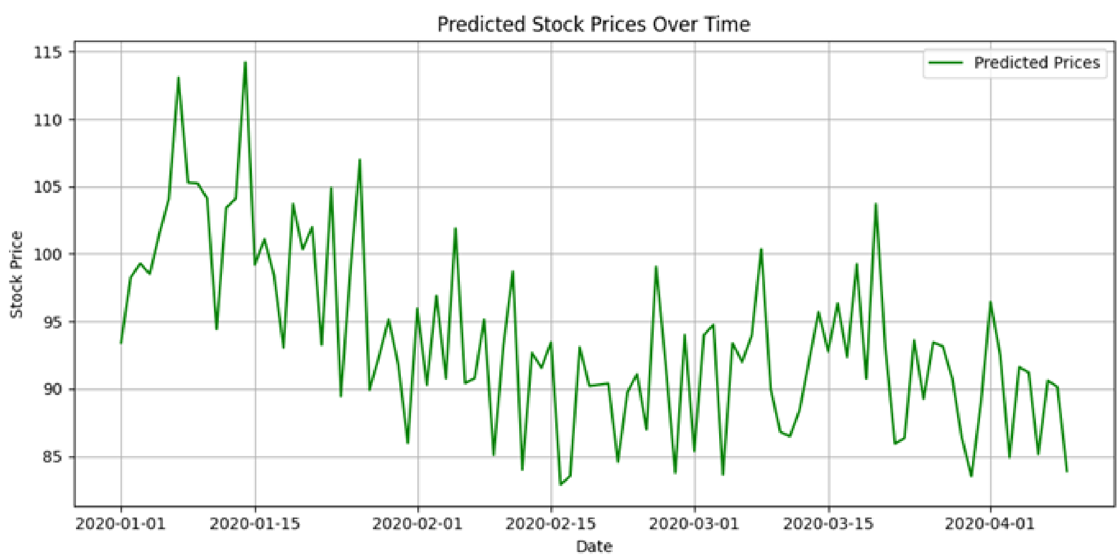

Figure 7: Predicted stock prices over time

The predictions of the stock prices with time (green line) are shown in figure 7, demonstrating the model predictions, trends, and fluctuations. The graph shows the transformation of the predictions to the changes in the market, which provides information about the consistency and reliability of the model to use in real-time financial analysis.

Table 1: Comparison results of the performance analysis

Methods	F1-Score (%)	Computational Time (s)
ANN	83.61	1.544
CNN	86.13	1.329
DNN	91.12	1.244
Proposed	97.16	1.150

Table 1 is a summary of the performance analysis of different methods using F1-Scores and computational time. ANN model obtained the F1-Score 83.61% and a time of computation 1.544 seconds. The CNN and DNN models had a better result of 86.13% and 91.12% of F1-Scores and computational times of 1.329 and 1.244 seconds. The accuracy and efficiency of the proposed method are the best, and it has the highest F1-Score of 97.16% and the lowest computational time of 1.150 seconds.

5 Discussion

Trade-off between bandwidth and latency of a network is essential to process financial stock data in real time, especially in an edge computing context. High bandwidth allows transfer of data at a higher speed and low latency allows timely decision making particularly in high frequency trading. The suggested approach can deal with these issues by applying wireless standards of communication with a low-latency, high-bandwidth data transfer. The data is processed at the edge devices to minimize the latency and congestion, and predictive analytics and machine learning control the bandwidth. The strategy will guarantee the efficient analysis of financial data in real time but will not overload the edge network. The edge devices based on wireless communication technology are essential in processing real-time financial stock data, as it makes it possible to process real-time data and reduce latency, as well as unnecessary cloud utilization. Storage optimization and bandwidth optimization are based on data compression and deduplication.

Edge-to-cloud synchronization is essential to transfer data efficiently and minimize the risk of data loss, which is critical to high-frequency trading. Privacy and security feature is also incorporated into the system, such as content-aware storage policies, encryption, and access control, to permit access only to sensitive financial information and allow its real-time analysis.

Ablation experiment determines the effectiveness of various elements of the suggested financial stock data processing approach. The Base Model has the lowest results in all the metrics. The addition of ATUKF enhances data cleaning as well as accuracy, but adds latency. FBPINN leads to increased precision and decreased latency, which is absent in the low-latency advantages of wireless communication. Wireless communication in itself enhances the efficiency of the data transfer systems, but it does not contribute significantly to the accuracy. The most accurate, the least latent, and the most efficient bandwidth consumption is shown to be carried out with the help of the entire model, including ATUKF, FBPINN, and wireless communication, which proves the importance of the combination of the above-mentioned techniques in the context of real-time processing of financial stock data.

6 Conclusion and Future Scope

The suggested model of real-time financial stock data processing in edge computing settings is much better at enhancing accuracy, precision, and computational efficiency. It uses an ATUKF to perform sound preprocessing and grouping and reveal concealed patterns in financial data. This is to guarantee quality input to be analyzed. FBPINN integration also increases the responsiveness and decreases the latency with an accuracy of 99.2 and a F1-score of 97.16. The proposed approach performs better on all metrics compared to traditional models such as ANN, CNN, and DNN, which puts the suggested approach in a better position to serve as an efficient solution to edge-based real-time analytics with respect to high-frequency financial data processing. Although the proposed method has an excellent performance, it has certain limitations. It is also sensitive to abrupt changes in data such as in some prediction deviations meaning it is sensitive to sharp spikes or noise in the data. Moreover, the existing implementation is tailored to the financial data of stocks, and its versatility to other areas needs to be further studied. The next step in the work should be the improvement of the model in the alternating edge conditions, the incorporation of a feedback mechanism in real time to adaptive learn and optimization of the system to accommodate the fusion of multi-source heterogeneous financial stock data. Besides, it would be helpful to test the system with real edge hardware to determine how effectively it can perform given resource limitations. This will assist in making sure that there is a scaled, feasible implementation of financial stock data processing in real life.

References

[1] Abdellatif, A. A., Samara, L., Mohamed, A., Erbad, A., Chiasserini, C. F., Guizani, M., ... & Laughton, J. (2021). Medge-chain: Leveraging edge computing and blockchain for efficient medical data exchange. *IEEE Internet of Things Journal*, *8*(21), 15762-15775. https://doi.org/10.1109/JIOT.2021.3052910

[2] Akbar, M., Ahmad, I., Mirza, M., Ali, M., & Barmavatu, P. (2024). Enhanced authentication for de-duplication of big data on cloud storage system using machine learning approach. *Cluster Computing*, *27*(3), 3683-3702. https://doi.org/10.1007/s10586-023-04171-y

[3] Aldhyani, T. H., & Alzahrani, A. (2022). Framework for predicting and modeling stock market prices based on deep learning algorithms. *Electronics*, *11*(19), 3149. https://doi.org/10.3390/electronics11193149

[4] Ansari, Y., Gillani, S., Bukhari, M., Lee, B., Maqsood, M., & Rho, S. (2024). A multifaceted approach to stock market trading using reinforcement learning. *IEEE Access*, *12*, 90041-90060. https://doi.org/10.1109/ACCESS.2024.3418510

[5] Bandi, A. (2021). Data streaming architecture for visualizing cryptocurrency temporal data. In *Computer Networks, Big Data and IoT: Proceedings of ICCBI 2020* (pp. 651-661). Singapore: Springer Singapore. https://doi.org/10.1007/978-981-16-0965-7_50

[6] Dautov, R., Distefano, S., Bruneo, D., Longo, F., Merlino, G., & Puliafito, A. (2021). Data agility through clustered edge computing and stream processing. *Concurrency and Computation: Practice and Experience*, *33*(7), 1-1. https://doi.org/10.1002/cpe.5093

[7] Egrioglu, E., Baş, E., & Chen, M. Y. (2022). Recurrent dendritic neuron model artificial neural network for time series forecasting. *Information Sciences*, *607*, 572-584. https://doi.org/10.1016/j.ins.2022.06.012

[8] Gao, R., Zhang, X., Zhang, H., Zhao, Q., & Wang, Y. (2022). Forecasting the overnight return direction of stock market index combining global market indices: A multiple-branch deep learning approach. *Expert Systems with Applications*, *194*, 116506. https://doi.org/10.1016/j.eswa.2022.116506

[9] Gülmez, B. (2023). Stock price prediction with optimized deep LSTM network with artificial rabbits optimization algorithm. *Expert Systems with Applications*, *227*, 120346. https://doi.org/10.1016/j.eswa.2023.120346

[10] Gunalan, K. K., Surya, & Chelvi, K.(2022). Stock Market Prediction. *International Academic Journal of Science and Engineering*, *9*(2), 18-22.

[11] Hou, D., Sun, Y., Dinavahi, V., & Wang, Y. (2024). Adaptive two-stage unscented Kalman filter for dynamic state estimation of synchronous generator under cyber attacks against measurements. *Journal of Modern Power Systems and Clean Energy*, *12*(5), 1408-1418. https://doi.org/10.35833/MPCE.2023.000352

[12] https://www.kaggle.com/datasets/jacksoncrow/stock-market-dataset

[13] Kumar, V. A., Kumar, A., Batth, R. S., Rashid, M., Gupta, S. K., & Raghuraman, M. (2021). Efficient data transfer in edge envisioned environment using artificial intelligence based edge node algorithm. *Transactions on Emerging Telecommunications Technologies*, *32*(6), e4110. https://doi.org/10.1002/ett.4110

[14] Li, M., Bi, X., Wang, L., & Han, X. (2021). A method of two-stage clustering learning based on improved DBSCAN and density peak algorithm. *Computer communications*, *167*, 75-84. https://doi.org/10.1016/j.comcom.2020.12.019

[15] Luo, Y. (2023). Financial data security management method and edge computing platform based on intelligent edge computing and big data. *IETE Journal of Research*, *69*(8), 5187-5195. https://doi.org/10.1080/03772063.2021.1973596

[16] Moseley, B., Markham, A., & Nissen-Meyer, T. (2023). Finite basis physics-informed neural networks (FBPINNs): a scalable domain decomposition approach for solving differential equations. *Advances in Computational Mathematics*, *49*(4), 62. https://doi.org/10.1007/s10444-023-10065-9

[17] Park, H. J., Kim, Y., & Kim, H. Y. (2022). Stock market forecasting using a multi-task approach integrating long short-term memory and the random forest framework. *Applied Soft Computing*, *114*, 108106. https://doi.org/10.1016/j.asoc.2021.108106

[18] Sankaranarayanan, S., Rodrigues, J. J., Sugumaran, V., & Kozlov, S. (2020). Data flow and distributed deep neural network based low latency IoT-edge computation model for big data environment. *Engineering Applications of Artificial Intelligence*, *94*, 103785. https://doi.org/10.1016/j.engappai.2020.103785

[19] Udayakumar, R., Anuradha, M., Gajmal, Y. M., & Elankavi, R. (2023). Anomaly detection for internet of things security attacks based on recent optimal federated deep learning model. *Journal of Internet Services and Information Security, 13*(3), 104-121. https://doi.org/10.58346/JISIS.2023.I3.007

[20] Wang, C., Chen, Y., Zhang, S., & Zhang, Q. (2022). Stock market index prediction using deep Transformer model. *Expert Systems with Applications*, *208*, 118128. https://doi.org/10.1016/j.eswa.2022.118128

[21] Wang, L., & Wang, Y. (2022). Supply chain financial service management system based on block chain IoT data sharing and edge computing. *Alexandria engineering journal*, *61*(1), 147-158. https://doi.org/10.1016/j.aej.2021.04.079

[22] Yemunarane, K., Chandramowleeswaran, G., Subramani, K., ALkhayyat, A., & Srinivas, G. (2024). Development and Management of E-Commerce Information Systems Using Edge Computing and Neural Networks. *Indian Journal of Information Sources and Services, 14*(2), 153–159. https://doi.org/10.51983/ijiss-2024.14.2.22

[23] Zeng, H. (2022). Influences of mobile edge computing-based service preloading on the early-warning of financial risks. *The Journal of Supercomputing*, *78*(9), 11621-11639. https://doi.org/10.1007/s11227-022-04329-2

Authors Biography

Neetu Venugopal Pillai, She works as a Lecturer in the Department of Computing and Engineering at the University of West London, RAK, UAE, and she is also a Research Scholar at Pillai College of Engineering, India. She has 9 years of teaching experience. Her research interests encompass cutting-edge areas such as Edge Computing, Cyber Security, Digital Forensics, and Data Mining.

Dr. Prashant Premji Nitnaware, He works as a Professor in the Department of Computer Engineering at Pillai College of Engineering, India. His areas of research interest are Artificial Intelligence, User Experience Design, Data Analytics, Blockchain Technology, and Machine Learning.