IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 3, 2021, accepted December 30, 2021, date of publication January 6, 2022, date of current version January 20, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3140646

Prediction of Network Traffic in Wireless Mesh Networks Using Hybrid Deep Learning Model

SMITA MAHAJAN[1], HARIKRISHNAN R.[2], AND KETAN KOTECHA[3]

[1]Department of Computer Science Engineering, Symbiosis Institute of Technology, Symbiosis International (Deemed) University, Pune, Maharashtra 412115, India
[2]Department of Electronics and Telecommunication Engineering, Symbiosis Institute of Technology, Symbiosis International (Deemed) University, Pune, Maharashtra 412115, India
[3]Department of Symbiosis Centre for Applied AI (SCAAI), Symbiosis Institute of Technology, Symbiosis International (Deemed) University, Pune, Maharashtra 412115, India

Corresponding author: Harikrishnan R. (dr.rhareish@gmail.com)

ABSTRACT Wireless mesh networks are getting adopted in the domain of network communication. Their main benefits include adaptability, configuration, and flexibility, with added efficiency in cost and transmission time. Traffic prediction refers to forecasting the traffic volumes in a network. The traffic volume includes incoming requests and outgoing data transmitted by the network nodes. The previous logs of traffic in the network are used for extracting patterns that help for accurate predictions. In this paper, an analysis of various existing traffic prediction methods is done. Specifically, the analysis of a case study where the performance of the High-Speed Diesel (HSD) pump is predicted by observing its output. A network of sensors form a less mesh network; sensors act as nodes while reading the parameters, namely, three-phase Current, Voltage, Temperature, and Vibration. In this case study, a High-Speed Diesel pumps' performance is predicted by predicting the vibration parameter as the output parameter. Other parameters affecting the performance of the High-Speed Diesel pump which are causing the change in vibration value are identified. Various algorithms, including Statistical Auto-Regressive Integration and Moving Average, Poisson's regression, and a few Machine Learning and Deep Learning algorithms like Decision Tree Regressor, Multi-Layer Perceptron, Linear Regression, and Long Short-Term Memory are implemented and evaluated for this purpose. Along with the comparison, a novel architecture using Convolution Neural Network and Long Short-Term Memory is described in this paper. The result and comparison between these give the clear understanding that the suggested novel Convo-LSTM model gives better performance and helps to predict the performance of the High-Speed Diesel pump. The proposed system makes a strong case for the network traffic prediction, where the use of historical data is collected over the wireless mesh network. A similar analogy can be used where this model could be implemented further for network monitoring tasks.

INDEX TERMS Deep learning, machine learning, multivariate time series analysis, prediction, wireless mesh networks.

I. INTRODUCTION

Networks are playing an important role during this age of digital expansion. For a given network, the most critical issues are its security, load balancing ability, maintainability, and speed. Various network topologies have existed, including bus, ring, star, mesh, hybrid, etc. Out of this, mesh networks have been one of the most popular choices owing to their stronger connection ability, lesser disadvantages in terms of lag and rigidity [1]. Wireless mesh networks are wireless based on the mesh topology, as shown in Figure 1. The main advantages of wireless mesh networks are their easy adaptability and configuration ability. Any future changes can be easily accommodated, thus leading to lower costs and maintenance. The main concepts related to a wireless mesh network are traffic prediction, traffic routing, and traffic control. Out of this, traffic prediction is a crucial aspect owing to being the fundamental block on which the performance of routing and congestion control algorithms is dependent. Traffic prediction refers to accurately predicting the possible traffic in a network at a given instance based on previous network data. An accurate estimate of the network traffic can help the network administrator improve the availability and transmission speeds of the network [2], [3].

The associate editor coordinating the review of this manuscript and approving it for publication was Pasquale De Meo.

Previous approaches for network traffic prediction have primarily focused on the host server logs along with consideration of the network parameter configuration [4]–[6]. This paper compares the performance of the six algorithms, namely, Decision Tree Regressor, Linear Regression, Multilayer Perceptron, Poisson's Regression, Auto Regressive Integration and Moving average, Long Short Term Memory. Out of these, ARIMA is a well-known model [7], Poisson's regression is a probabilistic model [8] and both are implemented traditionally in various applications. Decision Tree Regressor [9] and Linear Regressor [10] are machine learning algorithms, whereas Multi-Layer Perceptron [11]is a subset of Deep Neural Network and LSTM is an artificial recurrent neural network (RNN) architecture [12]. This way, the paper compares the performance of various different algorithms implemented on the same real-world data. This paper proposes a neoteric, hybrid technique for network traffic prediction in wireless mesh networks by focusing on the historical data collected over the network. Specifically, the case study of High-Speed Diesel (HSD) pumps is considered, where different sensors are used to collect the data values. Specifically, the case study of HSD pumps is considered, whose sensor readings are used to evaluate the performance. The installation of sensors(for reading various parameters) forms a mesh network that would be a good indicator to portray a typical mesh network traffic scenario. The sensor's mesh network is used to collect the data that includes readings of input and output parameters. In wireless mesh networks, the mesh nodes like in MANET nodes can form spontaneous connections with other nodes due to their intrinsic features to connect with and can traverse the network, collecting data from sensors, RFID-enabled nodes, and other fixed Wireless nodes [13]. Wireless Sensor Networks(WSN)s have been identified as a significant enabler of the IoT models since their inception. In IoT, all sensor nodes can get connected to the Internet to share and receive data [14]. A set of statistical and machine learning algorithms are used for the multivariate time series analysis of the collected data and subsequent output prediction. The traditional, fundamental, technical analysis approach is used by both statistical and non-statistical methods, with a significant focus on lag, first-order difference, and second-order difference. The stationarity and nature of the time series are more important in the fundamental analysis method [15]. Convolution Neural Network (CNN) is commonly employed in feature engineering because it focuses on the most evident elements in the line of sight. Long Short Term (LSTM) is extensively utilized in time series because it has the property of adopting/enhancing following the sequence of time [3].

In this paper, a novel Convo-LSTM architecture is proposed for the wireless mesh network's traffic prediction, citing a case study of the HSD pump. A Vibration forecasting model based on one dimensional (1-D) CNN-LSTM is built considering the properties of CNN and LSTM. The data tuples are collected over one year's period, with constant monitoring of the network, which is real-world data. The fundamental structure of the model is a hybrid or mixing of one dimensional (1-D) Convolution neural network and Long Short Term (LSTM). It has an input layer, one-dimensional convolution layer, pooling layer, LSTM hidden layer, and full connection layer in its architecture. The proposed model is evaluated in terms of Mean Square Error, Mean Absolute Error and Root Mean Square Error to check its performance.

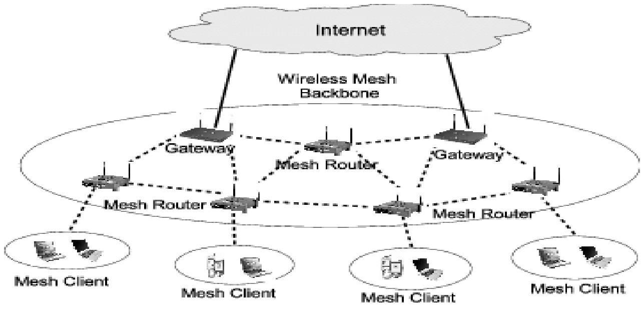


FIGURE 1. A sample wireless mesh network.

The main contributions through this paper could be enlisted as follows:

1) A hybrid model for network traffic prediction is proposed for the wireless mesh networks formed by various sensors, with the HSD pump as a case study.
2) A set of statistical, non-statistical, deep learning and machine learning algorithms are implemented, and results are compared for the collected multivariate time series data.
3) After applying these time-series based algorithms, unbiased analysis of the performances is done, which can be helpful for the researchers in the domain of wireless mesh networks.
4) An unbiased analysis of the results that a researcher can expect when applying these multivariate time-series algorithms in the domain of wireless mesh networks.

The paper outline is as follows: Section 2 provides a review of the previous work done in this domain. Section 3 explains the data collection process and data description. Section 4 elucidates the various analysis methods that are applied for traffic prediction. The obtained results are presented and analyzed in Section 5, while the conclusion of our findings is shown in Section 6.

II. BACKGROUND WORK

Network traffic prediction and routing have been a topic of interest and research in the last few decades. Approaches in this domain have comprised various time series models, namely machine learning algorithms, deep learning (neural networks), and various traditional statistical methods. Time series models like ARIMA have been preferred for forecasting, even in the case of regular networks. Zhou *et al.* [5] used a combination of ARIMA and GARCH models for prediction and modeling of the network traffic. Wavelet-based transformation models have also been used for this cause. Unlike image or text data problems, neural networks made inroads into network analysis and prediction almost a decade earlier. Khotanzad *et al.* [16] were one of the first to use neural networks for high-speed network traffic prediction. Alarcon *et al.* [17] applied a multi-resolution neural network for the task. Chen *et al.* [18] deployed a flexible neural tree for the resolution of traffic in the case of small scale networks. There have also been extensive studies comparing the performance of traditional methods versus deep learning networks [19]. Vinayakumar *et al.* [20] applied various sequence models, including Long Short Term Memory (LSTM)s, Recurrent neural networks (RNN), and Image Recognition Neural Networks (IRNN), for prediction of network traffic. Similar types of traditional and advanced approaches have also been applied for traffic prediction in the specific case of wireless networks. Amongst one of the earlier approaches, Gowrishankar and Satyanarayana [21] presented a neural network architecture for wireless network traffic prediction. Xiang *et al.* [22] had proposed a hybrid ANN-based approach for this task. Nikravesh *et al.* [23] analyzed the use of multiple techniques, including SVM, MLP, and MLPWD. Ke Wang, *et al.* [10] have suggested a hybrid model using CNN and LSTM where the ability of CNN and LSTM combined for traffic flow prediction. Stefany Coxe *et al.* [24] have worked on Poisson's regression and alternatives where they have stated that Count data represent the number of times an activity occurred over a specific period. e.g., suppose one wants to measure or observe how many hyper aggressive actions are expressed by children while playing during a playground area on particular occasions. In that case, that is nothing but count data. As per them, almost all Poisson regression models provide an easy method to implement analyses of count data. Farhan Mohammad Khan and Rajiv Gupta used an Auto-Regressive Integrated Moving Average (ARIMA) model to compare the accuracy of the predicted model [7]with a nonlinear autoregressive (NAR) neural network. Recently, for the daily prediction of COVID-19 cases for the next 50 days, the model was developed and implemented. Nie *et al.* [25], [26] have made multiple contributions in this domain in recent years. One of their initial approaches involved the use of deep belief networks for wireless mesh backbone networks. Recently, they further enhanced their work by applying reinforcement learning in an IoT setup [26]. Qiu *et al.* [27] deployed RNNs in a Spatio-temporal sense for improved performance for traffic prediction. Xu *et al.* [28] applied a multi-layer Gaussian framework for this task. Recently, researchers have also seen the rise of attention mechanism [29] and deep learning in interdisciplinary ways [30]. This trend is slowly reflected in traffic prediction.He *et al.* [31] applied a meta-learning scheme for faster traffic prediction in smaller networks. Li *et al.* [32] combined wavelet analysis with backpropagation neural networks for traffic flow analysis in wireless networks. Zhang *et al.* [33] considered a spatiotemporal network with a modified sequence model. Kim came up with an INGARCH model, an enhancement over the previous GARCH models [34]. Researchers have also used some statistical and evolutionary algorithms. Recently, most notably Li *et al.* [35] applied a Gaussian regressor with a Prophet model for user traffic prediction in networks. A quantum-PSO approach has also been tried out in this domain [36]. Costa *et al.*, give a thorough overview of predictive maintenance projects in Industry 4.0, identifying and classifying techniques, standards, and applications.Their survey's key contributions include a discussion of the existing issues and limitations in predictive maintenance, as well as a proposal for new taxonomy to define this research topic in light of Industry 4.0 requirements [37]. As per Costa *et al.*, the industry has entered an era due to the necessity to adapt and adopt new technologies. The Internet of Things (IoT) is a recent era for communication, in which all kinds of objects in our daily lives, such as smartphones, sensors, or devices, that have been linked to network-enabled objects (such as RFID) to communicate with each other and get to be a part of the Internet. Industry 4.0 is characterized by connectivity, data volume, tech gadgets, inventory reduction, customization, and controlled production [37]. Tan *et al.*, in their research, have analyzed recent advancements in smart monitoring and data analytics that have enabled infrastructure predictive maintenance (PdM). As per them, the industry is currently hesitant in adopting new smart monitoring sensors, information technologies, and data analytics to achieve PdM. PdM is data-driven, relying on smart monitoring and data analytics insights to prevent downtime through maintenance, protection, and repairs. PdM is a relatively new trend in the industry that has recently taken a leap in the industrial world since the 1990s; yet, their recent analysis of the industry revealed that its applicability in infrastructure maintenance is quite limited [38]. Chuang *et al.*, while emphasizing the importance of predictive maintenance, have stated that [39], business in all industries can be redefined with the emergence of AI and IoT. The information gathered is utilized not only used to draw inferences from the past but also to forecast the future. Artificial neural networks and evolutionary algorithms are two of the most common AI techniques for machine diagnosis. According to them, Predictive maintenance cuts down accidental device downtime, lowers maintenance costs, and extends equipment life cycle, among other benefits. The fundamental infrastructure of an IoT framework consists of sensors, actuators, computation servers, and the communication network [40]. The TCP/IP (Transmission Control Protocol/Internet Protocol) communication protocol transmits sensor data. The environmental sensing sensors are programmed into the programmable interface controller, and the data is saved in a historical manner [37], [39]. Pallavi *et al.* have mentioned that because IoT devices are typically located in geographically separated places, they communicate primarily over wireless mediums. They also have stated that Wireless channels are known for having significant distortion levels while being unstable. Communication techniques are essential for the analysis of IoT devices. In this scenario, reliably transferring data without too many retransmissions is a significant concern. The fundamental infrastructure of an IoT framework consists of sensors, actuators, computation servers, and the communication network [40]. MANET is a vital element of the IoT network, serving as its backbone. MANET nodes with mesh architecture can form spontaneous connections with other nodes due to their intrinsic features, requiring minimal infrastructure. MANET nodes can traverse the IoT network, collecting data from sensors, RFID-enabled nodes, and other fixed Wireless nodes. The MANET nodes take the most effective route to connect with the Internet gateways, which one is available. MANET nodes, like sensor nodes, can be employed as an essential technology in a variety of IoT applications.MANET nodes and sensor nodes (including RFID-enabled devices), forming a MESH Network, can be deployed in huge numbers due to their self-configuring nature. For the past decade, researchers have been working on the Internet of Things (IoT) using a variety of mature technologies such as Radio Frequency Identifiers (RFID), Wireless Sensor Network (WSN), Mobile Adhoc Network (MANET), and so on [41]. Nagarajan *et al.* have proposed remote health monitoring and data analysis by combining IoT and Deep Learning techniques. They have suggested a new IOT-based FoG-assisted cloud network architecture that collects real-time health care data from patients via numerous medical IoT sensor networks. The analyses of it by using a deep learning algorithm installed at a Fog-based Healthcare Platform. Furthermore, they have proposed a methodology used to analyze the process in real-time for smart cities. As per them, for timely, accurate and secure data analysis, new IOT-based FoG-assisted cloud network architecture can be effectuated to various domains such as traffic analysis and management, agriculture and smart farming, weather forecasting etc [14]. In the view of Manuel *et al.*, WSNs have been identified as a significant enabler of the IoT models since their inception. In IoT, all sensor nodes can get connected to the Internet to share and receive data; however, in WSNs, the nodes do not have a direct internet connection. To connect to the Internet, all nodes in the WSN need a mediator [42]. In their work, Krishnasamy *et al.* mention that a wireless sensor network (WSN) comprises a large number of sensor nodes that can both sense and communicate. The sensor nodes work together to gather and send the data to the sink node, also known as the coordinator node. The primary goal of sensor nodes is to oversee the environment before processing and transferring data to an analysis centre. Sensors are installed in locations, which are frequently uneven in design. Sensors are also installed randomly in specific sites that are irregular in shape, relying on the transmission range. As a result, an algorithm that can adapt to each geographic region with different deployment structures is required [43]. Authors Deverajan Ganesh Gopal *et al.* have researched a DANET, or dynamic ad hoc network, is a network of multiple dynamic nodes that do not necessitate any infrastructure. As and when needed, the movable nodes build a temporary network. This form of a network without the concept of a centralized entity, so nodes need to lean on another node to send packets. Multiple copies of data are created to enhance data availability. The access frequency and node level are taken into account while allocating copies [44]. In the research by Devarajan, Ganesh Gopal *et al.*, the Internet of Things (IoT)is a computational concept that envisions widespread Internet connectivity, transforming everyday objects into connected devices. The fundamental methodology in an IOT-based model is the transmission of billions or perhaps trillions of sensitive data capable of detecting the surrounding situation, communicating and transferring precise information, and then providing feedback to nature. Remote connections are frequently used to meet the adaptability and versatility required by IoT interchanges. While cellular innovations such as 3/4/5G provide interface separations of large devices, they necessitate framework support and legally allowed band. They have explained the concepts of IoT and IIoT, as well as the current trend of robotization and data exchange in manufacturing breakthroughs known as Industry 4.0 [45]. As per Farrukh *et al.*, it is understood that, With the creation of highly accurate and accurate algorithms, the investigation aims to focus on building more rigorous and practical methodologies [46].

It should be noted here that rarely has the work on traffic prediction focused on the historical data, which eventually can be used for traffic prediction. As a result, having models that are both resilient and appropriate is critical. Further, a standard paper highlighting all the contemporary machine learning and deep learning methods together could benefit young researchers in this domain. Also, there is hardly any notable work done where data is collected over a given period and analyzed for identifying the patterns in the collected data where real-life data is considered and eventually predicting the performance of the network. All these points highlight the scope for improvement and the need for our research work.

III. DATA COLLECTION AND PREPROCESSING

Before describing the algorithms used for modeling the data, the data collection and preparation process is as mentioned below.

A. DATA COLLECTION

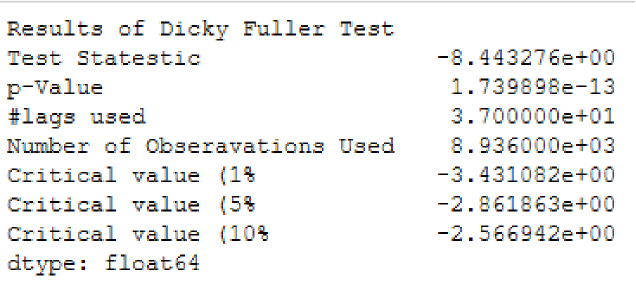
The data was collected for over a year using a set of HSD pumps. A total of eight sensors were placed for collecting data, forming a wireless mesh network. The collected data consists of seven input variables and one output variable. Each of these parameters are described in Table 1. The total data consists of 8960 such tuples, each containing one identifier (Date / Time), seven input variables(sensor reading for 3phase Current, 3phase Voltage and Temperature), and one output variable that is reading from the Vibration sensor. The data is divided into training and test sets in the ratio of 4:1, i.e. an 80% −20 % split are observed. The data is processed before converting it into a more algorithm-friendly format.

B. DATA PREPROCESSING

The following checks were performed on the data as a part of preprocessing:

TABLE 1. Description of the collected data parameters.

Parameter	Description
Date / Timestamp	The unique identifier indicating the timeframe when the data points were collected
Three Phase Current: IR, IY, IB	The three parameters corresponding to the three phase current in the system
Three Phase Voltage: VR, VY, VB	The three parameters corresponding to the three phase voltage in the system
Temperature	Temperature of the sensor of the HSD pump
Vibration	The output parameter corresponding to the traffic in the network



```
Results of Dicky Fuller Test
Test Statestic                       -8.443276e+00
p-Value                               1.739898e-13
#lags used                            3.700000e+01
Number of Obseravations Used          8.936000e+03
Critical value (1%                   -3.431082e+00
Critical value (5%                   -2.861863e+00
Critical value (10%                  -2.566942e+00
dtype: float64
```

FIGURE 2. ADfuller Test Results.

1) Check the time series stationarity, that is, if the time series is stationary or not.
2) Check whether a consistent mean and standard deviation exists in the collected data range or not. This is verified by plotting the mean and standard deviation values on a rolling window across the entire data range.
3) The ADFuller test for stationarity checks: The ADFuller test is used to check how well a trend persists over the time series. This is achieved by keeping a null hypothesis and an alternative hypothesis. Results of the test are shown in Figure 2. Our null hypothesis is that the time series has a common root and is non-stationary. The alternative hypothesis would be the series being stationary. The $p - value$ for the null hypothesis is calculated. The threshold is set to 0.05 for the p-value. As the value is observed to be less than the threshold, it can be concluded that the null hypothesis is true and that the series is indeed stationary.
4) Application of seasonal decompose: The seasonal decompose method is applied to get the triad values used for setting up the stationary time series that is used for forecasting. Residuals, Seasonality, and Trends are the three values from this method Figure 3.

IV. ALGORITHMS USED FOR TRAFFIC PREDICTION

In general, there are two types of traffic modelling for short-term traffic prediction: parametric and non-parametric methods. To employ parametric methods, first, a well-structured but flexible family of models is created, after which the model parameters using training data must be estimated. Forecasts can then be made using the model. This is a well-known method in traffic modelling, as seen in Figure 4. The Auto-Regressive Integrated Moving Average is a prominent modelling tool for traffic forecasting (ARIMA). [47]. Artificial intelligence-based regression models can provide the necessary skills. Regression is a solution for creating models capable of predicting the value of an output variable in accordance with a set of input variables that are widely utilized in many disciplines. AI-based algorithms are frequently utilized for complex regression models. This type of regression approach detects complex correlations between input variables and interactions between input variables and output variables automatically [48]. Thus, forecasting Internet traffic is critical for network planning, resource allocation and network anomaly detection caused by attacks. This is because enhanced TCP/IP (Transmission Control Protocol/Internet Protocol)traffic forecasting can assist network providers in optimizing their resources. Better traffic predictions can assist avoid congestion and resource waste in bandwidth allocation schemes. Short-term prediction and long-term prediction are the two categories of network traffic prediction. Short-term predicts traffic conditions in the near future based on historical and present traffic data. A forecast's horizon is only a few minutes long. On the other hand, long-term prediction provides traffic estimates for longer time periods, such as years. Traditional forecasting models such as the Poisson regression model (PRM) is used to model a counting variable, which is usually computed by using the maximum likelihood estimation (MLE) method [49]. Autoregressive (AR) and Autoregressive Integrated Moving Average (ARIMA) can figure out the linear and Short Range Dependencies (SRD) between terms, but not the Long Range Dependencies (LRD), resulting in poor performance when used for Internet traffic forecasting. Nonetheless, they are widely used [50].

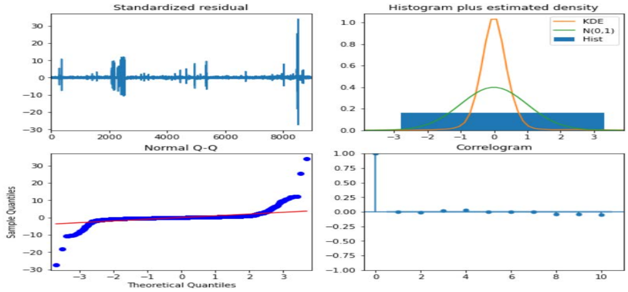


FIGURE 3. The seasonal decompose.

In this paper, both statistical and non-statistical algorithms are implemented for the task of output prediction. The description of each of these is given below:

A. DECISION TREE REGRESSOR

The Decision Tree Regressor(DTR) takes together all the input features and iteratively generates multiple trees trying out possible combinations of the root, internal, and leaf nodes amongst all the features [9]. The tree with the closest

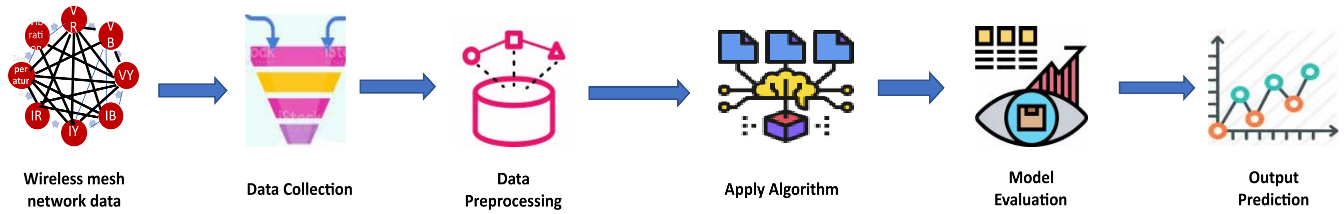


FIGURE 4. Block diagram for Prediction model.

predicted output to the actual traffic is considered the best tree for subsequent inference.

For each tree, for each node, two metrics can be calculated: Gini index and entropy. Gini index is a probabilistic measure that indicates the probability that the particular feature being at the given node would lead to the prediction error crossing a set threshold. At a particular node index n, the Gini index for a feature f_i is calculated as follows:

$$Gini\ Index_n = 1 - \Sigma_{i=1}^{n} p^2(f_i) \tag{1}$$

where $p(f_i)$ indicates the probability of f_i being present at node n. Ideally, the feature that produces the lowest Gini index for the particular node is assigned to that node. Overall, creating the decision tree would be to reduce the entropy, i.e. degree of randomness of possible traffic at each position. Entropy is defined as follows:

$$E(S) = \Sigma_{i=1}^{n} - p_i log_2 p_i \tag{2}$$

The lesser the entropy, the more confident and accurate the predictions would be. The Decision Tree Regressor figures out the best tree setup by iteratively trying out multiple combinations of positions of the seed and internal nodes in the tree.

B. LINEAR REGRESSION

The Linear Regression model uses a linear mapping of features to get a continuous prediction output. It is one of the most primitive algorithms represented mathematically as:

$$H(x) = \sum_{i=1}^{n} w_i X_i + \epsilon \tag{3}$$

where ϵ is the error factor put in to accommodate normalization [10].To approximate the given data, the regression and log-linear models can be employed.The data is modelled to match a straight line in (Simple) linear regression. A dependent variable, y i.e. response variable, can, for example, be described as a linear function of another random variable, x. i.e. predictor variable [10].

$$y = wx + b \tag{4}$$

C. MULTI-LAYER PERCEPTRON

The multi-layer perceptron model is a type of deep learning architecture where there is a combination of hidden layers through which the input features are passed to derive the output value. Each node in the hidden layer assigns an importance weight to each input that it receives from its preceding layer along with a bias value for error normalization. Generally, a fully connected setup is used wherein each node in a layer is connected to all nodes in its next layer [11].

Mathematically, the value at a particular node i in a given layer l is derived as follows:

$$H(x)_i^l = \sum_{j=1}^{n} w_j H(x)_j^{l-1} + b \tag{5}$$

A rectified linear unit activation function is used at the output layer to get a continuous prediction value. After comparing the predicted value with the actual value, the network back-propagates i.e. updates the initially randomized weights so that the predicted output matches the actual value as closely as possible.

$$w = w - \alpha dw \tag{6}$$

D. POISSON REGRESSION

The Poisson regression is a probabilistic model [8].It deploys a probabilistic mass function (PMF) to check what could be the probability of observing a particular continuous output y for a given input containing a set of dependent variables $X \in [X_1, X_2, .., X_n]$. This function is defined as follows:

$$PMF(y_i|x_i) = \frac{e^{-\lambda_i} * \lambda_i^{y_i}}{y_i!} \tag{7}$$

where λ_i is the mean rate, also meant to be the predicted value. The predicted regression output for a given input x is defined as follows:

$$\lambda_i = e^{x_i \beta} \tag{8}$$

Here, β is the regression coefficient or feature importance that is given to each dependent variable. The training objective of a Poisson regressor model is to find this β value. The best β value would be the one that produces the maximum value for PMF. The maximum value would be when the slope of the PMF curve is minimum, which is better derived by taking derivatives of the logarithm of PMF. This derivative equation is as follows:

$$ln(PMF) = \sum_{x=1}^{n} (y_i x_i \beta - e^{x_i \beta} - ln y_i!) \tag{9}$$

This is equated to zero to find the best β value. This value is then used to derive the prediction for a new test input x_p:

$$y_p = \lambda_p = e^{x_p \beta} \tag{10}$$

The training of Poisson regression is done with the objective of finding the values of the regression coefficient β that would make the vector of observed count y most likely. Following are the steps to be taken:

1) Convert the data set into only numeric values.
2) The data set should contain only non-negative integer values that represent the frequency of an event during a set interval. For our problem statement, it would be the traffic of network crossing host in particular interval.
3) Then find the regression variables that will influence the observed counts to derive the maximum PMF value.

E. AUTO REGRESSIVE INTEGRATION AND MOVING AVERAGE

ARIMA, that stands for Auto Regressive Integrated Moving Average, deploys a combination of auto-regression and moving average algorithms to get future predictions from past time series value [7]. Mathematically, it is represented as follows:

$$h_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \ldots + \beta_p Y_{t-p} \epsilon_t \tag{11}$$
$$y_t = h_t + \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + \ldots + \phi_q \epsilon_{t-q} \tag{12}$$

There are some terms based on auto-regression and some terms based on moving average. If terms in the time series are under-different, add more AR terms, and in cases of excess difference, then add more MA terms.ARIMA (p, d, q) method applies lag at the 1st or 2nd level if the non-stationary problem exists in the data; otherwise, if stationary without lag, then ARMA (p, q) is an alternative method; hence p for Moving Average (MA) and q for Autoregressive (AR) order that is the number of errors lag in ARIMA model forecast. The most common method used for making a sequence stationary is to subtract the initial value from the current value. Depending upon if the type of time series, i.e. univariate or multivariate, one or more lag is anticipated. Subsequently, the value of d signifies the smallest number of differentiation which is prescribed to keep the series stationary, so if without differentiation, the data series is still stationary, then d = 0. The identification method began by measuring the presence of autocorrelation (ACF) and partial autocorrelation (PACF) by plotting the correlogram by [51]. Then, depending on the ACF and PACF of the series, estimate of relevant models, setting the level of auto-regressive and moving averages. The (p, q) identified and the best model is performed based on the spikes and curve in the graph of ACF and PACF. Once the best model is selected, forecasting is done using parameters (p,d,q) given by the model. Diagnostic forecasting evaluation involves evaluating the efficacy of the currently built model using statistically relevant measures such as the Akaike information criterion (AIC), Bayesian criterion (BIC), and mean square error measurement [51].

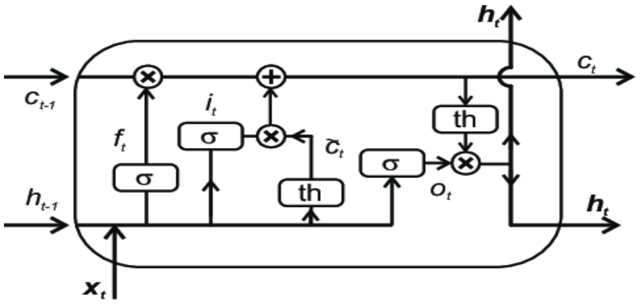


FIGURE 5. **Block diagram of LSTM architecture.**

F. LONG SHORT TERM MEMORY

Long Short-Term Memory is a recurrent network-based architecture where it keeps track of a cell state to remember certain memory trends in the series, shown in Figure 5. For every point in the state, the model decides whether to let go of some information, update some pattern information, or output any new information.LSTMs are specifically developed to prevent the problem of long-term dependency. All recurrent neural networks are made up of a series of repeated neural network modules. This recurring module in standard Recurrent Neural Networks (RNNs), shown in Figure 5, will have a simple structure, such as a single tanh layer [12].

There are three gates: forget, update, and output gates that operate on the given input for a time series input X_i and intermediate output h_t.

$$f_t = \sigma(\alpha_f x_t + \beta_f h_{t-1}) \tag{13}$$
$$o_t = \sigma(\alpha_o x_t + \beta_o h_{t-1}) \tag{14}$$

G. CONVO-LSTM

This paper proposes a novel combination of CNN along with LSTM such that the feature extraction ability of CNN can benefit the sequence recurrence mapping ability of the Recurrent Neural Networks(RNN). Figure 4 depicts the model structural diagram whereas Figure 6 depicts the proposed model's architectural diagram. An input layer, one-dimensional convolution layer, pooling layer, LSTM hidden layer, and full connection layer are the main constituents that build the main structure of Convo-LSTM. Lecun *et al.* proposed the CNN network model in 1998. The convolution operation extracts the attributes from the input layer vectors [52]. In this case, exclusively one dimensional - 1 D operations are only performed owing to the data structure. Pooling layers are deployed to reduce storage requirements and avoid the huge training costs in the system. The pooling layer subdivides the convolutional layer's small rectangular chunks to generate a single output from each block. Pooling can be done in various ways, such as by calculating the average or the maximum. The average pooling takes the average value of the block it is pooling, whereas the max-pooling takes the maximum of the block it is pooling [53]. Firstly, the CNN layer extracts the features from the data, which are the readings from the Current, Voltage, Temperature and Vibration sensor's readings collected over the previous year. The LSTM is then used to forecast the output, Vibration, based on the retrieved feature data. As per the experiment's findings, With the maximum prediction accuracy, the CNN-LSTM that is Convo-LSTM, can provide credible forecasting of the output parameter (Vibration).

1) TIME COMPLEXITY OF CONVO-LSTM

To determine the time complexity of both the forward propagation and backpropagation processes, the total number of operations at each 1D CNN layer must first be determined, and then the entire number of operations must be aggregated to determine the overall time complexity [54]. During forward propagation P, the number of connections to the preceding layer at a CNN layer,l, is N^{l-1} N^{l} the previous layer's number of connections is N^{l-1} N^{l}, an individual linear convolution, which is a linear weighted sum, is evaluated. Let S^{l-1} and W^{l-1}represent the vector sizes of the preceding layer output, S_{k}^{l-1} and the kernel (weight), respectively. A linear convolution is constituted of$(S^{l-1}W^{l-1})^{2}$ multiplications and S^{l-1} additions from a single connection, ignoring the boundary conditions. If the bias is ignored, the aggregate number of multiplications and additions in layer l will be:

$$N(mul)^{l} = N^{l-1}N^{l} * S^{l} - 1 * (W^{l-1})^{2} \tag{15}$$

$$N(add)^{l} = N^{l-1}N^{l} * S^{l-1} \tag{16}$$

Low computational complexity is attained in all of the 1D CNN. Thus, in forward propagation, the total number of multiplications T(mul) and the total number of addition T(add), in the CNN layer l will be

$$T_{\mathrm{FP}}(mul) = \sum_{l=0}^{L} N^{l-1}N^{l} * S^{l} - 1 * (W^{l-1})^{2} \tag{17}$$

$$T_{\mathrm{FP}}(add) = \sum_{l=0}^{L} N^{l-1}N^{l} * S^{l} - 1 \tag{18}$$

Now, similarly, at backpropagation iteration, the total number of multiplications and additions due to the first convolution will, therefore, be:

$$T_{\mathrm{BP}}(mul) = \sum_{l=0}^{L} N^{l+1}N^{l} * S^{l} + 1 * (W^{l+1})^{2} \tag{19}$$

$$T_{\mathrm{BP}}(add) = \sum_{l=0}^{L} N^{l+1}N^{l} * S^{l} + 1 \tag{20}$$

So at each BP iteration, the total number of multiplications and additions will be, respectively:

$$T_{\mathrm{FP}}(mul) + T_{\mathrm{FP}}(add) + T_{\mathrm{BP}}(mul)T_{\mathrm{BP}}(add) \tag{21}$$

2) STATISTICAL ANALYSIS

CNN stands for convolutional neural network and is a type of feed-forward neural network. It can be exploited to forecast time series with great success. Two inherent features, namely, Local perception and weight sharing, of CNN can significantly lower the number of parameters, which in turn help enhance model learning efficiency. The convolution layer and the pooling layer are the two fundamental constituents of CNN. Each convolution layer has a number of convolution kernels. The formula mentioned below is used for calculating them

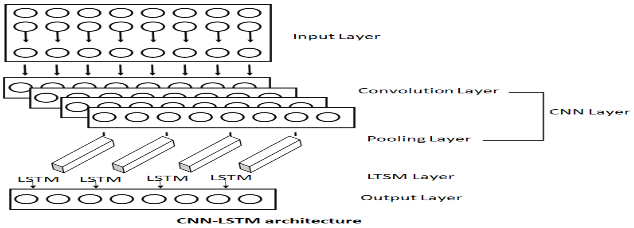


FIGURE 6. Proposed Convo-LSTM architecture.

$$I_{t} = \tanh(x_{t} * k_{t} + b_{t}) \tag{22}$$

where I_t is output value as a result of convolution, x_t is input vector,tanh is activation function, $b_t$is bias and k_t is convolution kernels' weight.The data features are obtained once the convolution layer completes the convolution operation, but, as the extracted feature dimensions are quite large, after the convolution layer, to lower the feature dimension and to lower the cost of training the network, a pooling layer is added. The forget gate receives the output value of the previous moment and the input value of the current time, with which the forget gate's output value is calculated, [52] as indicated in the following formula:

$$f_{t} = \sigma(W_{f}.[h_{t-1}, X_{t}] + b_{f}) \tag{23}$$

The last time's output value and the current time's input value are both fed into the input gate, and the output value and candidate cell state of the input gate are calculated, as shown in the formulas below:

$$i_{t} = \sigma(W_{i}.[h_{t-1}, X_{t}] + b_{f}) \tag{24}$$

The final output O_t is calculated as follows:

$$o_{t} = \sigma(\alpha_{o}x_{t} + \beta_{o}h_{t-1}) \tag{25}$$

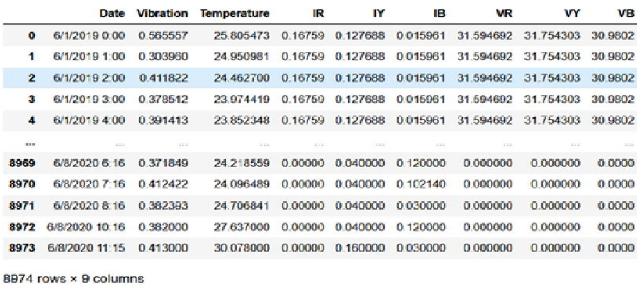
$$h_{t} = o_{t} * \tanh(C_{t}) \tag{26}$$

where f_t is having the value range of(0,1),W_f and b_f are the weight and bias of forget gate. Similarly,W_i and b_i are the weight and bias of the input gate having value range (0,1)and C_t is the output of the current cell with value and (0,1) [52].

The following is a summary of the proposed architecture: The convolution layer and the pooling layer are the two fundamental components of CNN. Each convolution layer has several convolution kernels. The data features are extracted after the convolution operation of the Complexity the convolution layer. As the pulled feature dimensions are very large, a pooling layer is added after the convolution layer to reduce the feature dimension and to reduce the cost of training the network. The convolution operation extracts the features from

TABLE 2. Summary of proposed Convo-LSTM architecture.

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 15, 64)	2304
max_pooling 1d (MaxPooling1D)	(None, 14, 64)	0
conv1d_1 (Conv1D)	(None, 14, 32)	6176
max_pooling1d_1 (MaxPooling1)	(None, 13, 32)	0
LSTM (LSTM)	(None, 13, 128)	82432
LSTM_1 (LSTM)	(None, 13, 128)	131584
flatten (Flatten)	(None, 1664)	0
dense (Dense)	(None, 64)	106560
dropout (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 16)	1040
dropout_1 (Dropout)	(None, 16)	0
dense_2 (Dense)	(None, 1)	17



	Date	Vibration	Temperature	IR	IY	IB	VR	VY	VB
0	6/1/2019 0:00	0.565557	25.805473	0.16759	0.127688	0.015961	31.594692	31.754303	30.9802
1	6/1/2019 1:00	0.303960	24.950981	0.16759	0.127688	0.015961	31.594692	31.754303	30.9802
2	6/1/2019 2:00	0.411822	24.462700	0.16759	0.127688	0.015961	31.594692	31.754303	30.9802
3	6/1/2019 3:00	0.378512	23.974419	0.16759	0.127688	0.015961	31.594692	31.754303	30.9802
4	6/1/2019 4:00	0.391413	23.852348	0.16759	0.127688	0.015961	31.594692	31.754303	30.9802
...	...	...	...	...	...	...	...	...	...
8969	6/8/2020 6:16	0.371849	24.218559	0.00000	0.040000	0.120000	0.000000	0.000000	0.0000
8970	6/8/2020 7:16	0.412422	24.096489	0.00000	0.040000	0.102140	0.000000	0.000000	0.0000
8971	6/8/2020 8:16	0.382393	24.706841	0.00000	0.040000	0.030000	0.000000	0.000000	0.0000
8972	6/8/2020 10:16	0.382000	27.637000	0.00000	0.040000	0.120000	0.000000	0.000000	0.0000
8973	6/8/2020 11:15	0.413000	30.078000	0.00000	0.160000	0.030000	0.000000	0.000000	0.0000

8974 rows × 9 columns

FIGURE 7. Data set-snapshot.

the input layer vectors. In this case, exclusively 1D operations are only performed owing to the structure of the data. To avoid the introduction of huge training costs in the system, pooling layers are also deployed to reduce storage requirements. The summary of the proposed architecture is shown in Table 1 To demonstrate the model's usefulness, the performance of ARIMA, Decision Tree Regressor, Linear Regressor, Multi-Layer Perceptron, and Long Short Term Memory is compared in this work, using the same training and test sets (sample mentioned in Figure 7 under the same operating environment. All the model development and its performance evaluations are performed under the running environment of Intel (R) Core(TM) i5-7300HQ CPU @ 2.50GHz 2.50 GHz, 8.00 GB (7.87 GB usable) RAM and Windows10. The sensors used for current, temperature and vibration include the LM35 temperature sensor (Texas Instruments, Dallas, TX, USA) is a precision IC temperature sensor with a proportional output (in C) is used for temperature. A Hall Effect-based DC current sensor, Allegro MicroSystems LLC's ACS712 current transducer, is used in this circuit (Worcester, MA, USA). High-frequency accelerometers with a flat frequency response up to 28kHz for multi-stage compressors and boiler feed pumps monitoring and bearing wear detection.

3) SAMPLE DATA SET DESCRIPTION

The proposed system makes a strong case for the network traffic prediction, where the use of historical(time series data) data is collected over the wireless mesh network. The sensors nodes with mesh architecture form connections. The data is collected over one year, starting from 1st June 2019 to 8th June 2020 are obtained from the sensors measuring the Temperature, Three-phase Voltage, Three-phase current and Vibration values of the HSD Pump. A snapshot of the data is shown in Figure 7. As to implement various algorithms, the data is required to be divided into the training set, testing data set and validation set. The first 7178 readings of the data are taken as the training set, and the data of 1345 readings are taken as the validation set the last 450 readings as the test set. According to the influence factors, including the Temperature, Three Phase Current (IR, IY, IB), Three Phase voltage (VR, VY, VB), the HSD pump's Vibration is predicted On daily, hourly, weekly, monthly, and yearly basis. As per standard industry practice, the LM35 temperature sensor (Texas Instruments, Dallas, TX, USA) is a precision IC temperature sensor with a proportional output is used for temperature. A Hall Effect-based DC current sensor is the ideal method of measurement for monitoring the current of the motor. Allegro MicroSystems LLC's ACS712 current transducer is used in this circuit (Worcester, MA, USA) [41]. High-frequency accelerometers with a flat frequency response up to 28kHz for multi-stage compressors and boiler feed pumps monitoring and bearing wear detection. The snapshot of the data set is shown in Figure 7. **Model Description** Various parameters of the Convo-LSTM are included in Table 2, which shows the CNN-LSTM parameter settings used in this experiment. The specific model is built as follows, based on the parameter settings of the Convo-LSTM network: A three-dimensional data vector is used as the input training data (None, 10, 7), where 10 is the time step size and as there are seven attributes of the input data. A one-dimensional convolution layer is used to send the data at first, which extracts additional features and produces a three-dimensional output vector (None, 15, 64), where the size of the convolution layer filters is 64. After passing through the pooling layer, the vector is transformed into a three-dimensional output vector (None, 13, 32). The output vector is then trained using the LSTM layer and two dense layers, and the output data (None, 64) goes through another complete connection layer after training to retrieve the output value; 64 is the number of hidden units in the LSTM layer. This CNN-LSTM model structure is shown in Figure 6.

V. RESULTS

The models are trained using the data from the processed training set; namely, Decision Tree Regressor (DTR), Linear Regression (LR), Multi-layer Perceptron (MLP), Poisson Regression, Auto-Regressive Integrated Moving Average (ARIMA), Long Short-Term Memory (LSTM), Convo-LSTM respectively are trained. To forecast the output from the test set data, this fully trained model is used. The result of the predictions of the model is compared with the actual value of the output from the data set. Among the six forecasting

TABLE 3. Results for hourly prediction model.

Approach	MAE	MSE	RMSE
DTR	0.12	0.47	0.21
Linear Regression	0.13	0.42	0.17
Multi-layer Perceptron	0.16	0.70	0.49
Poisson Regression	0.06	0.50	0.68
ARIMA	0.19	0.46	0.67
LSTM	0.46	0.24	0.49
Convo-LSTM	0.1	0.025	0.16

TABLE 4. Results for daily prediction model.

Approach	MAE	MSE	RMSE
DTR	0.1199	0.4612	0.2127
Linear Regression	0.1253	0.4161	0.1732
Multi-layer Perceptron	0.1608	0.6941	0.4818
Poisson Regression	0.055	0.464	0.681
ARIMA	0.15	0.46	0.68
LSTM	0.46	0.23	0.48
Convo-LSTM	0.1	0.025	0.159

TABLE 5. Results for weekly prediction model.

Approach	MAE	MSE	RMSE
DTR	0.13	0.47	0.22
Linear Regression	0.14	0.48	0.18
Multi-layer Perceptron	0.16	0.69	0.48
Poisson Regression	0.05	0.46	0.68
ARIMA	0.15	0.49	0.677
LSTM	0.45	0.23	0.49
Convo-LSTM	0.1	0.025	0.159

TABLE 6. Results for monthly prediction model.

Approach	MAE	MSE	RMSE
DTR	0.11	0.46	0.21
Linear Regression	0.12	0.41	0.17
Multi-layer Perceptron	0.16	0.69	0.48
Poisson Regression	0.055	0.464	0.681
ARIMA	0.14	0.45	0.67
LSTM	0.463	0.239	0.489
Convo-LSTM	0.1	0.025	0.159

methods, Decision Tree Regressor (DTR), Linear Regression (LR), Multi-layer Perceptron (MLP), Poisson Regression, Auto-Regressive Integrated Moving Average (ARIMA), Long Short-Term Memory (LSTM), and Convo-LSTM, the maximum degree of broken line fitting is shown by Convo-LSTM, which practically fits with each other and the MLP model shows the lowest degree of broken line fitting.

The aforementioned model is evaluated across multiple time intervals:

TABLE 7. Results for yearly prediction model.

Approach	MAE	MSE	RMSE
DTR	0.1199	0.4612	0.2127
Linear Regression	0.1253	0.4161	0.1732
Multi-layer Perceptron	0.1608	0.6941	0.4818
Poisson Regression	0.055	0.464	0.681
ARIMA	0.148	0.458	0.677
LSTM	0.463	0.239	0.489
Convo-LSTM	0.1	0.025	0.159

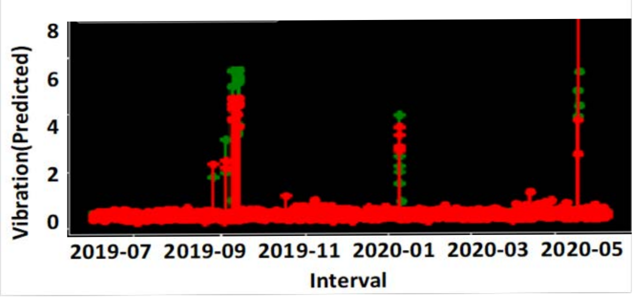


FIGURE 8. Actual vs Predicted vibrations using Poisson's algorithm.

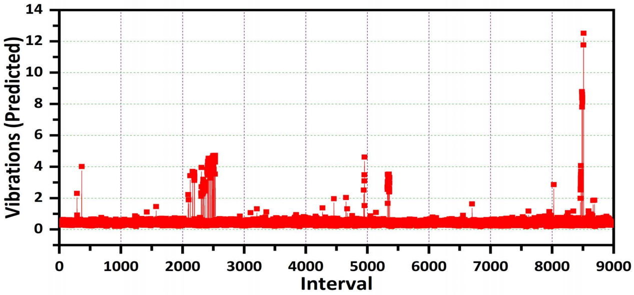


FIGURE 9. Actual vs Predicted vibrations using ARIMA algorithm.

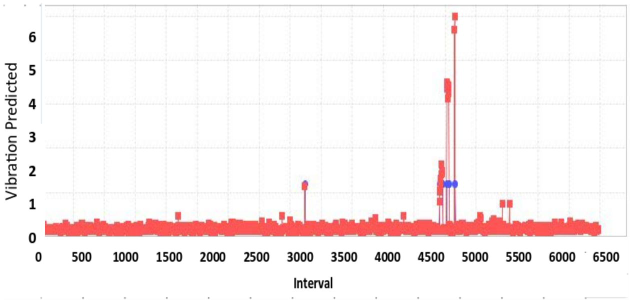


FIGURE 10. Actual vs Predicted vibrations using DTR algorithm.

- Hourly: The Sensor readings are logged at the interval of every hour.
- Daily: The Sensor readings are logged as the cumulative value for the entire day.
- Weekly: The Sensor readings are logged at the weekly level, thereby negating any daily variations observed between any specific days.

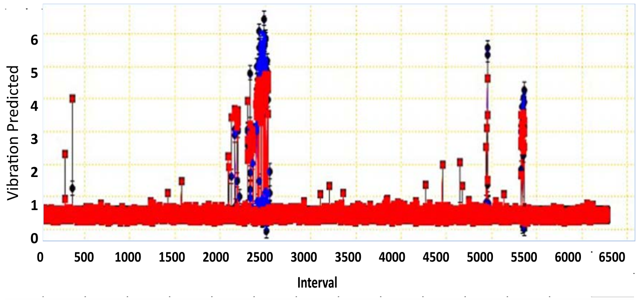


FIGURE 11. Actual vs Predicted vibrations using Linear Regression algorithm.

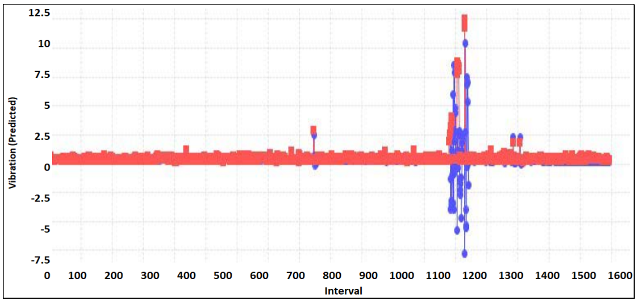


FIGURE 12. Actual vs Predicted vibrations using MLP algorithm.

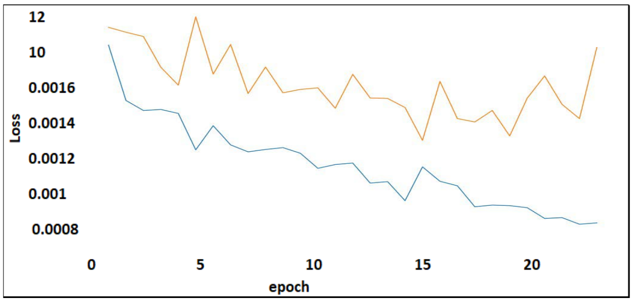


FIGURE 13. Actual vs Predicted vibrations using LSTM algorithm.

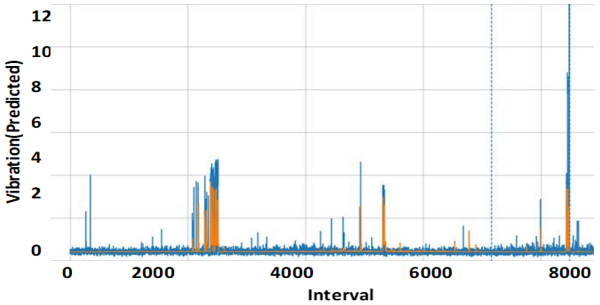


FIGURE 14. Actual vs Predicted vibrations using Convo-LSTM algorithm.

- Monthly: The Sensor readings are logged on a month over month basis.
- Yearly: The Sensor readings on a yearly basis.

The results are evaluated across the following three metrics:

- Mean Absolute Errors (MAE): Absolute difference between actual and predicted output value.From the observations it is found that the MAE for Convo-LSTM model is 0.1, which indicates the model's performance over other models.

$$\sum_{i=1}^{D} |x_i - y_i| \tag{27}$$

- Mean Square Error (MSE): Square of the difference between actual and predicted output.

$$\sum_{i=1}^{D} (x_i - y_i)^2 \tag{28}$$

- Root Mean Squared Error (RMSE): Square root of the MSE value. The result shows the comparison of the RMSE value of Convo-LSTM over other models.

$$\sqrt{\frac{1}{n}\Sigma_{i=1}^{n}\left(\frac{d_i - f_i}{\sigma_i}\right)^2} \tag{29}$$

The hourly, daily, weekly, monthly and yearly prediction of the output is presented as the results as mentioned in Table 3 to Table 7. The result shows the comparison of the MAE, MSE and RMSE value of Convo-LSTM over other models for hourly, daily, weekly, monthly, and yearly models, which indicates the model's performance over other models. From the observations, it is found that the MAE for the Convo-LSTM model is 0.1, MSE is 0.025, and RMSE is 0.16, which indicates the model's performance over other models. For interpretability of the results,the graphs indicating comparison between predicted and actual output for the implemented algorithms are shown,where Figure 8 MLP, Figure 9 ARIMA, Figure 10 DTR, Figure 11 Linear Regression, Figure 12 Multi-Layer Perceptron, Figure 13 LSTM, Figure 14 Convo-LSTM. The resultant graphs have been added to this paper.

VI. CONCLUSION

This paper proposes a hybrid model for traffic prediction in wireless mesh networks by application of regression methods on system configuration parameters. Specifically, six different algorithms are applied: decision tree regressor, linear regression, multi-layer perceptron, Poisson regression, ARIMA, and LSTM on the three main feature types: three-phase current, three-phase voltage, and temperature to predict the output- Vibration of the HSD Pump. This paper also proposes a new Convo-LSTM setup for this task and achieve good results from the same. The system was evaluated on five different intervals: hourly, daily, weekly, monthly, and yearly and it is found the Convo-LSTM algorithm to be the best performing one. A time-series multivariate data set is similar to that of wireless mesh networks data set. This gives a direction to the researchers to implement the proposed algorithm to predict the volume of network traffic. Future work in the domain includes the application of some contemporary methods like the attention mechanism and transformers for traffic prediction. Multimodal networks that combine the physical configuration values and the network system log values can also be proposed. As network systems become larger and more distributed in nature, smart algorithms that can automatically predict the incoming traffic and accordingly allocate the resources would become the need of the hour.

REFERENCES

[1] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: A survey," *IEEE Wireless Commun.*, vol. 11, no. 6, pp. 6–28, Dec. 2004.

[2] A. Cilfone, L. Davoli, L. Belli, and G. Ferrari, "Wireless mesh networking: An IoT-oriented perspective survey on relevant technologies," *Future Internet*, vol. 11, no. 4, p. 99, Apr. 2019.

[3] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2224–2287, 3rd Quart., 2019.

[4] H. Moayedi and M. Masnadi-Shirazi, "Arima model for network traffic prediction and anomaly detection," in *Proc. Int. Symp. Inf. Technol.*, vol. 4, Sep. 2008, pp. 1–6.

[5] B. Zhou, D. He, Z. Sun, and W. H. Ng, "Network traffic modeling and prediction with arima/garch," in *Proc. HET-NETs Conf.*, 2005, pp. 1–10.

[6] S. Han-Lin, J. Yue-Hui, C. Yi-Dong, and C. Shi-Duan, "Network traffic prediction by a wavelet-based combined model," *Chin. Phys. B*, vol. 18, no. 11, p. 4760, 2009.

[7] F. M. Khan and R. Gupta, "ARIMA and NAR based prediction model for time series analysis of COVID-19 cases in India," *J. Saf. Sci. Resilience*, vol. 1, no. 1, pp. 12–18, Sep. 2020.

[8] K. F. Sellers and B. Premeaux, "Conway–maxwell–Poisson regression models for dispersed count data," *WIREs Comput. Statist.*, vol. 13, no. 6, p. e1533, Nov. 2021.

[9] S. Walker, W. Khan, K. Katic, W. Maassen, and W. Zeiler, "Accuracy of different machine learning algorithms and added-value of predicting aggregated-level energy performance of commercial buildings," *Energy Buildings*, vol. 209, Jan. 2020, Art. no. 109705.

[10] K. Wang, C. Ma, Y. Qiao, X. Lu, W. Hao, and S. Dong, "A hybrid deep learning model with 1DCNN-LSTM-attention networks for short-term traffic flow prediction," *Phys. A, Stat. Mech. Appl.*, vol. 583, Jan. 2021, Art. no. 126293.

[11] Y. Liu, S. Liu, Y. Wang, F. Lombardi, and J. Han, "A stochastic computational multi-layer perceptron with backward propagation," *IEEE Trans. Comput.*, vol. 67, no. 9, pp. 1273–1286, Sep. 2018.

[12] Y. Yu, X. Si, C. Hu, and Z. Jianxun, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural Comput.*, vol. 31, no. 7, pp. 1235–1270, Jul. 2019.

[13] L. Nie, X. Wang, L. Wan, S. Yu, H. Song, and D. Jiang, "Network traffic prediction based on deep belief network and spatiotemporal compressive sensing in wireless mesh backbone networks," *Wireless Commun. Mobile Comput.*, vol. 2018, May 2018, Art. no. 1260860.

[14] S. M. Nagarajan, G. G. Deverajan, P. Chatterjee, W. Alnumay, and U. Ghosh, "Effective task scheduling algorithm with deep learning for Internet of Health Things (IoHT) in sustainable smart cities," *Sustain. Cities Soc.*, vol. 71, Jan. 2021, Art. no. 102945.

[15] B. Lindemann, T. Müller, H. Vietz, N. Jazdi, and M. Weyrich, "A survey on long short-term memory networks for time series prediction," *Proc. CIRP*, vol. 99, pp. 650–655, Jan. 2021.

[16] A. Khotanzad and N. Sadek, "Multi-scale high-speed network traffic prediction using combination of neural networks," in *Proc. Int. Joint Conf. Neural Netw.*, vol. 2, Jul. 2003, pp. 1071–1075.

[17] V. Alarcon-Aquino and J. A. Barria, "Multiresolution FIR neural-network-based learning algorithm applied to network traffic prediction," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 36, no. 2, pp. 208–220, Mar. 2006.

[18] Y. Chen, B. Yang, and Q. Meng, "Small-time scale network traffic prediction based on flexible neural tree," *Appl. Soft Comput.*, vol. 12, no. 1, pp. 274–279, Jan. 2012.

[19] T. P. Oliveira, J. S. Barbar, and A. S. Soares, "Computer network traffic prediction: A comparison between traditional and deep learning neural networks," *Int. J. Big Data Intell.*, vol. 3, no. 1, pp. 28–37, 2016.

[20] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Applying deep learning approaches for network traffic prediction," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Sep. 2017, pp. 2353–2358.

[21] Gowrishankar and P. S. Satyanarayana, "Neural network based traffic prediction for wireless data networks," *Int. J. Comput. Intell. Syst.*, vol. 1, no. 4, pp. 379–389, Dec. 2008.

[22] L. Xiang, X.-H. Ge, C. Liu, L. Shu, and C.-X. Wang, "A new hybrid network traffic prediction method," in *Proc. IEEE Global Telecommun. Conf.*, Dec. 2010, pp. 1–5.

[23] A. Y. Nikravesh, S. A. Ajila, C.-H. Lung, and W. Ding, "Mobile network traffic prediction using mlp, mlpwd, and SVM," in *Proc. IEEE Int. Congr. Big Data*, May 2016, pp. 402–409.

[24] S. Coxe, S. G. West, and L. S. Aiken, "The analysis of count data: A gentle introduction to Poisson regression and its alternatives," *J. Personality Assessment*, vol. 91, no. 2, pp. 121–136, Feb. 2009.

[25] L. Nie, D. Jiang, S. Yu, and H. Song, "Network traffic prediction based on deep belief network in wireless mesh backbone networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2017, pp. 1–5.

[26] L. Nie, Z. Ning, M. S. Obaidat, B. Sadoun, H. Wang, S. Li, L. Guo, and G. Wang, "A reinforcement learning-based network traffic prediction mechanism in intelligent Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 17, no. 3, pp. 2169–2180, Mar. 2021.

[27] C. Qiu, Y. Zhang, Z. Feng, P. Zhang, and S. Cui, "Spatio-temporal wireless traffic prediction with recurrent neural network," *IEEE Wireless Commun. Lett.*, vol. 7, no. 4, pp. 554–557, Aug. 2018.

[28] Y. Xu, F. Yin, W. Xu, J. Lin, and S. Cui, "Wireless traffic prediction with scalable Gaussian process: Framework, algorithms, and verification," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1291–1306, Jun. 2019.

[29] P. Ratadiya and D. Mishra, "An attention ensemble based approach for multilabel profanity detection," in *Proc. Int. Conf. Data Mining Workshops (ICDMW)*, Nov. 2019, pp. 544–550.

[30] P. Ratadiya, K. Asawa, and O. Nikhal, "A decentralized aggregation mechanism for training deep learning models using smart contract system for bank loan prediction," 2020, *arXiv:2011.10981*.

[31] Q. He, A. Moayyedi, G. Dan, G. P. Koudouridis, and P. Tengkvist, "A meta-learning scheme for adaptive short-term network traffic prediction," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 10, pp. 2271–2283, Oct. 2020.

[32] M. Li, Y. Wang, Z. Wang, and H. Zheng, "A deep learning method based on an attention mechanism for wireless network traffic prediction," *Ad Hoc Netw.*, vol. 107, Jun. 2020, Art. no. 102258.

[33] D. Zhang, L. Liu, C. Xie, B. Yang, and Q. Liu, "Citywide cellular traffic prediction based on a hybrid spatiotemporal network," *Algorithms*, vol. 13, no. 1, p. 20, Jan. 2020.

[34] M. Kim, "Network traffic prediction based on INGARCH model," *Wireless Netw.*, vol. 26, no. 8, pp. 6189–6202, Nov. 2020.

[35] Y. Li, Z. Ma, Z. Pan, N. Liu, and X. You, "Prophet model and Gaussian process regression based user traffic prediction in wireless networks," *Sci. China Inf. Sci.*, vol. 63, no. 4, pp. 1–8, Apr. 2020.

[36] Y. Li, J. Huang, and H. Chen, "Time series prediction of wireless network traffic flow based on wavelet analysis and BP neural network," *J. Phys., Conf. Ser.*, vol. 1533, no. 3, Apr. 2020, Art. no. 032098.

[37] T. Zonta, C. A. D. Costa, R. D. R. Righi, M. J. D. Lima, E. S. D. Trindade, and G. P. Li, "Predictive maintenance in the industry 4.0: A systematic literature review," *Comput. Ind. Eng.*, vol. 2020, Oct. 2020, Art. no. 106889.

[38] M. Tan, I. Ubhayaratne, Y. Huo, F. B. Varela, and Y. Xiang, "Predictive maintenance based on smart monitoring and data analytics," in *Proc. Australas. Corrosion Assoc. Corrosion Prevention Conf.*, 2019, pp. 1–10.

[39] S.-Y. Chuang, N. Sahoo, H.-W. Lin, and Y.-H. Chang, "Predictive maintenance with sensor data analytics on a raspberry pi-based experimental platform," *Sensors*, vol. 19, no. 18, p. 3884, Sep. 2019.

[40] P. Sethi and S. R. Sarangi, "Internet of Things: Architectures, protocols, and applications," *J. Electr. Comput. Eng.*, vol. 2017, Jan. 2017, Art. no. 9324035.

[41] S. Mukherjee and G. P. Biswas, "Networking for IoT and applications using existing communication technology," *Egyptian Informat. J.*, vol. 19, no. 2, pp. 107–127, Jul. 2018.

[42] A. J. Manuel, G. G. Deverajan, R. Patan, and A. H. Gandomi, "Optimization of routing-based clustering approaches in wireless sensor network: Review and open research issues," *Electronics*, vol. 9, no. 10, p. 1630, Oct. 2020.

[43] L. Krishnasamy, R. Dhanaraj, D. G. Gopal, T. R. Gadekallu, M. Aboudaif, and E. A. Nasr, "A heuristic angular clustering framework for secured statistical data aggregation in sensor networks," *Sensors*, vol. 20, no. 17, p. 4937, Aug. 2020.

[44] D. G. Gopal and R. Saravanan, ''Selfish node detection based on evidence by trust authority and selfish replica allocation in danet,'' *Int. J. Inf. Commun. Technol.*, vol. 9, no. 4, pp. 473–491, 2016.

[45] G. G. Deverajan, V. Muthukumaran, C.-H. Hsu, M. Karuppiah, Y.-C. Chung, and Y.-H. Chen, ''Public key encryption with equality test for industrial Internet of Things system in cloud computing,'' *Trans. Emerg. Telecommun. Technol.*, vol. 4, Jan. 2021, Art. no. e4202.

[46] Y. Ali Farrukh, I. Khan, Z. Ahmad, and R. M. Elavarasan, ''A sequential supervised machine learning approach for cyber attack detection in a smart grid system,'' 2021, *arXiv:2108.00476*.

[47] W. Alajali, W. Zhou, S. Wen, and Y. Wang, ''Intersection traffic prediction using decision tree models,'' *Symmetry*, vol. 10, no. 9, p. 386, Sep. 2018.

[48] J. P. T. Higgins and S. G. Thompson, ''Controlling the risk of spurious findings from meta-regression,'' *Statist. Med.*, vol. 23, no. 11, pp. 1663–1682, 2004.

[49] M. Amin, M. N. Akram, and M. Amanullah, ''On the James-Stein estimator for the Poisson regression model,'' in *Proc. Commun. Statistics-Simulation Comput.*, pp. 1–13, 2020.

[50] C. Katris and S. Daskalaki, ''Comparing forecasting approaches for internet traffic,'' *Expert Syst. Appl.*, vol. 42, no. 21, pp. 8172–8183, 2015.

[51] E. Elakkiya, M. Radha, and R. Sathy, ''Application of Arima model for predicting cashew nut production in India—An analysis,'' *Int. J. Res. Bus. Manage.*, vol. 5, pp. 45–52, 2017.

[52] W. Lu, J. Li, Y. Li, A. Sun, and J. Wang, ''A CNN-LSTM-based model to forecast stock prices,'' *Complexity*, vol. 2020, May 2020, Art. no. 6622927.

[53] M. Alawad and M. Lin, ''Stochastic-based deep convolutional networks with reconfigurable logic fabric,'' *IEEE Trans. Multi-Scale Comput. Syst.*, vol. 2, no. 4, pp. 242–256, Oct. 2016.

[54] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, ''1d convolutional neural networks and applications: A survey,'' *Mech. Syst. signal Process.*, vol. 151, May 2021, Art. no. 107398.

SMITA MAHAJAN received the bachelor's degree from Shivaji University, Maharashtra, India, and the master's degree in information technology from Mumbai University. She is currently pursuing the Ph.D. degree in computer science engineering with Symbiosis International University, Pune, Maharashtra. She is currently working as an Assistant Professor with the Computer Science Engineering Department, Symbiosis Institute of Technology, Symbiosis International Deemed University, Pune. Her main research interests include computer networking, wireless and broadband networks, deep learning, and machine learning.

HARIKRISHNAN R. received the bachelor's degree in electrical and electronics engineering from the University of Madras, the master's degree in energy system engineering from VIT University, Vellore, the master's degree in embedded system technologies from Anna University, Chennai, India, and the Ph.D. degree in electrical engineering from Sathyabama University, Chennai. He has 21 years of teaching, research, and industrial experience. He is currently working as an Associate Professor with the Electronics and Telecommunication Engineering Department, Symbiosis Institute of Technology, Symbiosis International Deemed University, Pune, India. His main research interests include smart grid, the Internet of Things, artificial intelligence, and wireless sensor networks.

KETAN KOTECHA received the M.Tech. and Ph.D. degrees from the IIT Bombay. He is currently the Head of the Symbiosis Centre for Applied AI (SCAAI). He has expertise and experience in cutting-edge research and projects in AI and deep learning for more than the last 25 years. He has published more than 100 widely in a number of excellent peer-reviewed journals on various topics ranging from cutting-edge AI, education policies, teaching-learning practices, and AI for all. He has published three patents and delivered keynote speeches at various national and international forums, including at Machine Intelligence Laboratory, USA, at IIT Bombay, under World Bank Project, at the International Indian Science Festival organized by the Department of Science Technology, Government of India, and many more. He was a recipient of the two SPARC projects worth INR 166 lacs from MHRD Government of India in AI in collaboration with Arizona State University, USA, and The University of Queensland, Australia, and also a recipient of numerous prestigious awards like Erasmus+ Faculty Mobility Grant to Poland, DUO-India Professors Fellowship for Research in Responsible AI in collaboration with Brunel University, U.K., LEAP Grant at Cambridge University, U.K., UKIERI Grant with Aston University, U.K., and a Grant from the Royal Academy of Engineering, U.K., under Newton Bhabha Fund. He is also an Associate Editor of IEEE Access journal.