**Web Traffic Time Series Forecasting**

**Abstract:**

These days, web traffic determining is a significant issue as this can make misfortunes the activities of major websites. Time-arrangement gauging has been an interesting issue for research. Foreseeing future time arrangement esteems is one of the most troublesome issues in the business. The time arrangement field incorporates various issues, running from derivation and investigation to estimating and arrangement. Determining the organization traffic and showing it in a dashboard that refreshes progressively would be the most effective approach to pass on the data. Making a Dashboard would help in checking and breaking down constant information. These days, we are excessively reliant on Google worker yet on the off chance that we need to have a worker for enormous clients we might have anticipated the quantity of clients from earlier years to dodge worker breakdown. Time Series anticipating is urgent to different spaces.

**Objectives:**

The dataset was examined and the example information utilized was 'India'. Further, the dataset was separated into preparing and testing sets. For the time arrangement, we plotted the number of hits versus days alongside genuine qualities and conjectures for the article 'India' during the testing time frame.

**Scope of the Project:**

A traffic the executive strategy or plan ought to be set up to diminish the danger of such disasters which could be negative to the presence of the organization. As of not long ago, there wasn't a requirement for such apparatuses as most workers could deal with the traffic deluge yet the cell phone age has expanded the request to a particularly level for certain websites that organizations couldn't have responded rapidly enough to keep up their ordinary client care level.

**Existing System:**

The most well-known model for linear univariate time series forecasting is the autoregressive integrated moving average (ARIMA), which encompasses other autoregressive time series models, including auto regression (AR), moving average (MA), and autoregressive moving average (ARMA). Additionally, linear support vector regression (SVR) treats the forecasting problem as a typical regression problem with time-varying parameters. However, these models are mostly limited to linear univariate time series and do not scale well to MTS. To forecast MTS data, vector autoregression (VAR), a generalization of AR-based models, was proposed. VAR is probably the most well-known model in MTS forecasting. Nevertheless, neither AR-based nor VAR-based models capture non-linearity. For that reason, substantial effort has been put into non-linear models for time series forecasting based on kernel methods, ensembles Gaussian processes or regime switching. Still, these approaches apply predetermined non-linearity’s and may fail to recognize different forms of non-linearity for different MTS. Recently, deep neural networks have received a great amount of attention due to their ability to capture non-linear interdependencies. Long shortterm memory (LSTM) variant of recurrent neural network, has shown promising results in several NLP tasks and has also been employed for MTS forecasting. Work in this area began with using naive RNN improved with hybrid models that combined ARIMA and multilayer perceptron’s, and then most recently progressed to dynamic Boltzmann machines with RNN. Although these models can be applied to MTS, they mainly target univariate or bivariate time series. To the best of our knowledge, the long- and short-term time-series network (LSTNet) is the first model designed specifically for MTS forecasting with up to hundreds of time series. LSTNet uses CNNs to capture short-term patterns, and LSTM or GRU for memorizing relatively long-term patterns. In practice, however, LSTM and GRU cannot memorize very long-term interdependencies due to training instability and the gradient vanishing problem. To address this, LSTNet adds either a recurrent-skip layer or a typical attention mechanism. Also part of the overall model is traditional autoregression, which helps to mitigate the scale insensitivity of neural networks.

**Disadvantage:**

* Long short-term memory (LSTM) variant of recurrent neural network, has shown promising results in several NLP tasks and has also been employed for MTS forecasting.

**Proposed System:**

Discrete wavelet Transform separates information signals into fundamental wavelet capacities. Since the time-arrangement information secured for examination is loud in nature it is vital to finish pre-handling of the information. Here the DWT parts down the information into bits of low recurrence. So for best outcomes, we can apply the calculation to the fragments. Since the information should be fixed in ARIMA hence we utilize high-recurrence information as a prescient commitment. RNN utilizes information from low frequencies as information. It was later seen that this method yields agreeable outcomes for less and more information that isn't the autonomously actualized circumstance for ARIMA and RNN.

**Advantage:**

* Determining an effective strategy for load balancing of web pages residing in the cloud
* Forecasting future trends based on historical data and
* Understanding the user behavior.

**LITERATURE SURVEY:**

**REFERENCE PAPER-1**

**TITLE:** Web Traffic Prediction of Wikipedia Pages

**AUTHOR:** Navyasree Petluri, Eyhab Al-Masri

**ABSTRACT:**

Applying a forecasting model for the purpose of predicting web traffic. In particular, we use existing Web Traffic Time Series Forecasting dataset by Google to predict future traffic of Wikipedia articles. Predicting web traffic can help web site owners in many ways including: (a) determining an effective strategy for load balancing of web pages residing in the cloud, (b) forecasting future trends based on historical data and (c) understanding the user behavior. To achieve the goals of this research work, we built a time-series model that utilizes RNN seq2seq model. We then investigate the use of symmetric mean absolute percentage error (SMAPE) for measuring the overall performance and accuracy of the developed model. Finally, we compare the outcome of our developed model to existing ones to determine the effectiveness of our proposed method in predicting future traffic of Wikipedia articles.

**ALGORITHM USED:**

* RNN seq2seq model.

**REFERENCE PAPER-2**

**TITLE:** Temporal Pattern Attention for Multivariate Time Series Forecasting

**AUTHOR:** Shun-Yao Shih\* · Fan-Keng Sun\* · Hung-yi Lee

**ABSTRACT:**

**Forecasting of multivariate time series data, for instance the prediction of electricity consumption, solar power production, and polyphonic piano pieces, has numerous valuable applications. However, complex and non-linear interdependencies between time steps and series complicate this task. To obtain accurate prediction, it is crucial to model long-term dependency in time series data, which can be achieved by recurrent neural networks (RNNs) with an attention mechanism. The typical attention mechanism reviews the information at each previous time step and selects relevant information to help generate the outputs; however, it fails to capture temporal patterns across multiple time steps. In this paper, we propose using a set of filters to extract time-invariant temporal patterns, similar to transforming time series data into its “frequency domain”. Then we propose a novel attention mechanism to select relevant time series, and use its frequency domain information for multivariate forecasting. We apply the proposed model on several real-world tasks and achieve state-of-the-art performance in almost all of cases.**

**ALGORITHM USED:**

* frequency domain

**REFERENCE PAPER-3**

**TITLE:** Connecting the Dots: Multivariate Time Series Forecasting with Graph Neural Networks

**AUTHOR:** Zonghan Wu, Shirui Pan∗ , Guodong Long, Jing Jiang

**ABSTRACT:**

Modeling multivariate time series has long been a subject that has attracted researchers from a diverse range of fields including economics, finance, and traffic. A basic assumption behind multivariate time series forecasting is that its variables depend on one another but, upon looking closely, it’s fair to say that existing methods fail to fully exploit latent spatial dependencies between pairs of variables. In recent years, meanwhile, graph neural networks (GNNs) have shown high capability in handling relational dependencies. GNNs require well-defined graph structures for information propagation which means they cannot be applied directly for multivariate time series where the dependencies are not known in advance. In this paper, we propose a general graph neural network framework designed specifically for multivariate time series data. Our approach automatically extracts the uni-directed relations among variables through a graph learning module, into which external knowledge like variable attributes can be easily integrated. A novel mix-hop propagation layer and a dilated inception layer are further proposed to capture the spatial and temporal dependencies within the time series

**ALGORITHM USED:**

* Graph neural networks

**REFERENCE PAPER-4**

**TITLE:** Time series forecasting using improved ARIMA

**AUTHOR**: Soheila Mehrmolaei, Mohammad Reza Keyvanpour

**ABSTRACT:**

Forecasting is one of the main goal's mining of time series databases. Time series forecasting has been shown effective in suitable decision making in various domains. So far, a variety of techniques have been proposed to obtain goal of prediction and analysis of literature this area is in different directions. In this regard, in this paper, there are two goals. First, provide a review. For this goal, this paper classifies previous major works that investigated the forecasting of time series data in different application areas. Second, propose a novel approach to improve ARIMA model by applying a mean of estimation error for time series forecasting. Experimental results indicate that the proposed approach can improve performance in the process of time series data forecasting.

**ALGORITHM USED:**

* Improved ARIMA

**REFERENCE PAPER-5**

**TITLE**: Predicting Computer Network Traffic: A Time Series ForecastingApproach Using DWT, ARIMA and RNN

**AUTHOR**: Rishabh Madan, Partha Sarathi Mangipudi

**ABSTRACT:**

This paper proposes the Discrete Wavelet Transform (DWT), Auto Regressive Integrated Moving Averages (ARIMA) model and Recurrent Neural Network (RNN) based technique for forecasting the computer network traffic. Computer network traffic is sampled on computer networking device connected to the internet. At first, discrete wavelet transform is used to decompose the traffic data into non-linear (approximate) and linear (detailed) components. After that, detailed and approximate components are reconstructed using inverse DWT and predictions are made using Auto Regressive Moving Average (ARIMA) and Recurrent Neural Networks (RNN), respectively. Internet traffic is a time series which can be used to predict the future traffic trends in a computer network. Numerous computer network management tasks depend heavily on the information about the network traffic. This forecasting is very useful for numerous applications, such as congestion control, anomaly detection, and bandwidth allocation. Our method is easy to implement and computationally less expensive which can be easily applied at the data centers, improving the quality of service (QoS) and reducing the cost.

**ALGORITHM USED:**

* DWT,ARIMA and RNN.

**REFERENCE PAPER-6**

**TITLE:** Time Series Forecasting Based on Complex Network Analysis

**AUTHOR:** Shengzhong Mao Fuyuan Xiao

**ABSTRACT:**

Time series forecasting, especially from the perspective of the network, has been a hot research topic. In this paper, based on the analysis of complex network, a novel method is proposed for more accurate time series predictions. First, time series data are mapped into a network by visibility graph. Then, the link prediction method is adopted to calculate the similarity index. Considering that node distance is an important factor in the network, we take that into account to determine the weight coefficients and improve the predictive results. To fully verify the validity of the proposed method, it is applied to some representative time series data sets with different characteristics. The data values are recorded daily, monthly, and yearly. The error measurement and correlation analysis show that our method has a good prediction performance. It is believed that this paper will not only contribute to time series forecasting in theory but also take effect in practice.

**ALGORITHM USED:**

* Complex Network Analysis.

**REFERENCE PAPER-7**

**TITLE:** Efficient Prediction of Network Traffic for Real-Time Applications

**AUTHOR**: Shengzhong Mao Fuyuan Xiao

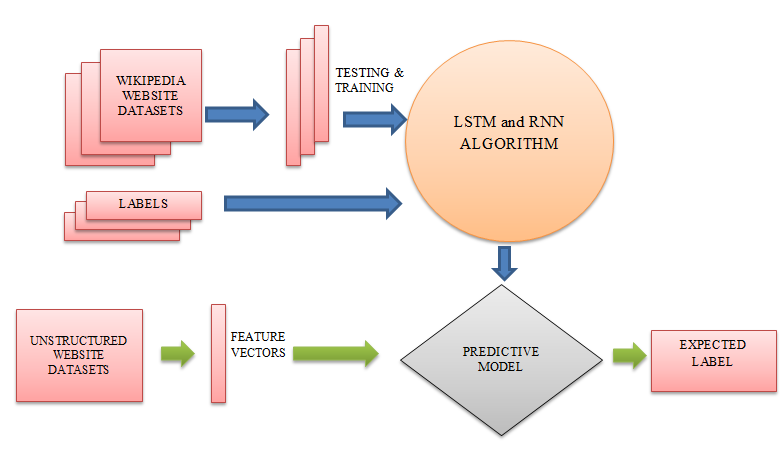
**ABSTRACT:**

Time series forecasting, especially from the perspective of the network, has been a hot research topic. In this paper, based on the analysis of complex network, a novel method is proposed for more accurate time series predictions. First, time series data are mapped into a network by visibility graph. Then, the link prediction method is adopted to calculate the similarity index. Considering that node distance is an important factor in the network, we take that into account to determine the weight coefficients and improve the predictive results. To fully verify the validity of the proposed method, it is applied to some representative time series data sets with different characteristics. The data values are recorded daily, monthly, and yearly. The error measurement and correlation analysis show that our method has a good prediction performance. It is believed that this paper will not only contribute to time series forecasting in theory but also take effect in practice.

**ALGORITHM USED:**

* Error Measurement and Correlation Analysis

**System Architecture Diagram:**



**SYSTEM CONFIGURATION:**

# H/W SYSTEM CONFIGURATION:

* Processor - Intel
* Speed - 1.1 GHz
* RAM - 4 Gb
* Hard Disk - 260 GB

# S/W SYSTEM CONFIGURATION:

* Operating System - Windows 7/8/10
* Front End - Html,Css
* Scripts - java/ Jsp
* Tool - Net Beans 7.3.1

**MODULES:**

* PRE-PROCESSING WEB TRAFFIC TIME SERIES DATASET
* IMPLEMENTATION OF ARIMA AND LSTM RNN.
* GETTING RESULTS AND ANALYSIS OF FORECASTING

**PRE-PROCESSING WEB TRAFFIC TIME SERIES DATASET:**

The core dataset used for this project is Wikipedia's Web Traffic Time Series Forecasting (provided via Kaggle) consisting of approximately 145,000 Wikipedia articles. The dataset includes a field representing the time series or multiple points given in an order of time. For example, each of the time series represents a number of daily views of a different Wikipedia article, Due to the fact that the does not differentiate between values of zero and missing values in the traffic data, this causes some ambiguity in the overall predictions which we will have to account for in our prediction model. Once the dataset is loaded, it always helps to view the dataset to ensure it has been read in correctly. We can see that the dataset consists of four columns: the type of event (“view”, “search” or “lead”) , the user id (unique identifier), timestamp (time and date when user accessed the website) and just the date. Time variables and unique identifiers are what I like to call “junk” variables. They do not assist the model in prediction. However, time is an important feature in our case as we want to know whether a user will become a lead the next day. An individual user id is also important. So, how can we still use this information. As a user may view or search a website one or more times a day, it makes sense to aggregate this information to reduce the number of records per user but also to be able to create a daily dataset. To predict whether a user becomes a lead the next day, we need to know whether the customer was a lead the previous day, whether they viewed and searched the website the previous day, and total number of views and searches to current date.

**IMPLEMENTATION OF ARIMA AND LSTM RNN.**

We analyzed existing winning model provided to the Kaggle’s competition which used RNN seq2seq model. This prediction model is built based on: a) number of hits, b) features which are extracted from page URLs, c) day of week - analyzes the weekly seasonality information, d) year-to-year autocorrelation (quarterly and yearly), e) page popularity and, f) lagged page views. We rebuilt the existing winning model with the entire dataset as a training data using RNN seq2seq model with the help of Encoder/decoder Architecture. Encoder is cuDNN GRU as it performs task with better speed compare to regular tensors. Decoder is TensorFlow GRUBlockCell. The generated results from decoding are used as inputs to the next step till the end of the sequence for a given batch size. While your model is training, you can test its accuracy using a method called cross***-***validation***.***In cross-validation, you run multiple different training test splits and then average the results, instead of relying entirely on a single particular training set. The most common type of cross-validation is k-fold cross-validation most commonly with K set to 5 or 10. For example, to do five-fold cross-validation, the original dataset is partitioned into five parts of equal or close to equal size. Each of these parts is called a “fold”. Then a series of five models is trained one per fold. The first model: Model one, is trained using folds 2 through 5 as the training set and evaluated using fold 1 as the test set. The second model: Model 2, is trained using Folds 1, 3, 4, and 5 as the training set, and evaluated using Fold 2 as the test set, and so on. When this process is done, we have five accuracy values, one per fold. It is common to take the average of the accuracy scores. You can set up your model for cross-fold validation as follows. Here we have specified five rounds of cross-validation, told the model to run 1000 iterations but to stop early if it no longer shows any improvement in accuracy. The final line tells us at which iteration did the model produce a tree with lowest error.

**GETTING RESULTS AND ANALYSIS OF FORECASTING:**

We analyzed existing winning model provided to the Kaggle’s competition which used RNN seq2seq model. This prediction model is built based on: a) number of hits, b) features which are extracted from page URLs, c) day of week - analyzes the weekly seasonality information, d) year-to-year autocorrelation (quarterly and yearly), e) page popularity and, f) lagged page views. We rebuilt the existing winning model with the entire dataset as a training data using RNN seq2seq model with the help of Encoder/decoder Architecture. Encoder is cuDNN GRU as it performs task with better speed compare to regular tensors. Decoder is TensorFlow GRUBlockCell. The generated results from decoding are used as inputs to the next step till the end of the sequence for a given batch size. We have run 3 models in parallel on 2 seeds. Using tensor board, we generated the resulting SMAPE graph for each of the three existing and new models to check the variations in the model performance at every step. With minimalistic inputs, our algorithm performs best in uncovering the features independently as they make use of sequential information where model is run for every element of the sample, with the current output relying on the stored past calculations. With simple median as feature, proposed system is able to estimate the proportion of web traffic with which it was able to do predictions with stability based on the previous computations which being the reason for slight improvement in case of rolling median and Fibonacci median as features.

**NON-FUNCTIONAL REQUIREMENTS:-**

**FEASIBILITY STUDY:**

**1. FEASIBILITY STUDY**

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

* ECONOMICAL FEASIBILITY
* TECHNICAL FEASIBILITY
* SOCIAL FEASIBILITY

**ECONOMICAL FEASIBILITY**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

**TECHNICAL FEASIBILITY**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

**SOCIAL FEASIBILITY:**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

**JAVA**

Java is a programming language originally developed by James Gosling at Sun Microsystems now a subsidiary of Oracle Corporation, and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. Java applications are typically compiled to bytecode (class file) that can run on any Java Virtual Machine (JVM) regardless of computer architecture. Java is general-purpose, concurrent, class-based, and object-oriented, and is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere".

The original and reference implementation Java compilers, virtual machines, and class libraries were developed by Sun from 1995. As of May 2007, in compliance with the specifications of the Java Community Process, Sun relicensed most of their Java technologies under the GNU General Public License. Others have also developed alternative implementations of these Sun technologies, such as the GNU Compiler for Java and GNU Classpath.

**Java Platform**

One characteristic of Java is portability, which means that computer programs written in the Java language must run similarly on any supported hardware/operating-system platform. This is achieved by compiling the Java language code to an intermediate representation called Java bytecode, instead of directly to platform-specific machine code. Java bytecode instructions are analogous to machine code, but are intended to be interpreted by a virtual machine (VM) written specifically for the host hardware. End-users commonly use a Java Runtime Environment (JRE) installed on their own machine for standalone Java applications, or in a Web browser for Java applets.

Standardized libraries provide a generic way to access host-specific features such as graphics, threading and networking.

A major benefit of using bytecode is porting. However, the overhead of interpretation means that interpreted programs almost always run more slowly than programs compiled to native executables would, and Java suffered a reputation for poor performance. This gap has been narrowed by a number of optimization techniques introduced in the more recent JVM implementations.

**Implementations**

Sun Microsystems officially licenses the Java Standard Edition platform for Linux, Mac OS X and Solaris. Although in the past Sun has licensed Java to Microsoft, the license has expired and has not been renewed. Through a network of third-party vendors and licensees, alternative Java environments are available for these and other platforms.

Sun's trademark license for usage of the Java brand insists that all implementations be "compatible". This resulted in a legal dispute with Microsoft after Sun claimed that the Microsoft implementation did not support RMI or JNI and had added platform-specific features of their own. Sun sued in 1997, and in 2001 won a settlement of $20 million as well as a court order enforcing the terms of the license from Sun. As a result, Microsoft no longer ships Java with Windows, and in recent versions of Windows, Internet Explorer cannot support Java applets without a third-party plugin. Sun, and others, has made available free Java run-time systems for those and other versions of Windows.

Platform-independent Java is essential to the Java EE strategy, and an even more rigorous validation is required to certify an implementation. This environment enables portable server-side applications, such as Web services, Java Servlets, and Enterprise JavaBeans, as well as with embedded systems based on OSGi, using Embedded Java environments. Through the new GlassFish project, Sun is working to create a fully functional, unified open source implementation of the Java EE technologies.

Sun also distributes a superset of the JRE called the Java Development Kit (commonly known as the JDK), which includes development tools such as the Java compiler, Javadoc, Jar and debugger.

**Performance**

Programs written in Java have a reputation for being slower and requiring more memory than those written in some other languages. However, Java programs' execution speed improved significantly with the introduction of Just-in-time compilation in 1997/1998 for Java 1.1, the addition of language features supporting better code analysis (such as inner classes, StringBuffer class, optional assertions, ect.), and optimizations in the Java Virtual Machine itself, such as HotSpot becoming the default for Sun's JVM in 2000.

To boost even further the speed performances that can be achieved using the Java language Systronix made JStik, a microcontroller based on the aJile Systems line of embedded Java processors.

**Automatic memory management**

Java uses an automatic garbage collector to manage memory in the object lifecycle. The programmer determines when objects are created, and the Java runtime is responsible for recovering the memory once objects are no longer in use. Once no references to an object remain, the unreachable memory becomes eligible to be freed automatically by the garbage collector. Something similar to a memory leak may still occur if a programmer's code holds a reference to an object that is no longer needed, typically when objects that are no longer needed are stored in containers that are still in use. If methods for a nonexistent object are called, a "null pointer exception" is thrown.

One of the ideas behind Java's automatic memory management model is that programmers be spared the burden of having to perform manual memory management. In some languages memory for the creation of objects is implicitly allocated on the stack, or explicitly allocated and deallocated from the heap. Either way, the responsibility of managing memory resides with the programmer. If the program does not deallocate an object, a memory leak occurs. If the program attempts to access or deallocate memory that has already been deallocated, the result is undefined and difficult to predict, and the program is likely to become unstable and/or crash. This can be partially remedied by the use of smart pointers, but these add overhead and complexity. Note that garbage collection does not prevent 'logical' memory leaks, i.e. those where the memory is still referenced but never used.

Garbage collection may happen at any time. Ideally, it will occur when a program is idle. It is guaranteed to be triggered if there is insufficient free memory on the heap to allocate a new object; this can cause a program to stall momentarily. Explicit memory management is not possible in Java.

Java does not support C/C++ style pointer arithmetic, where object addresses and unsigned integers (usually long integers) can be used interchangeably. This allows the garbage collector to relocate referenced objects, and ensures type safety and security.

As in C++ and some other object-oriented languages, variables of Java's primitive data types are not objects. Values of primitive types are either stored directly in fields (for objects) or on the stack (for methods) rather than on the heap, as commonly true for objects (but see Escape analysis). This was a conscious decision by Java's designers for performance reasons. Because of this, Java was not considered to be a pure object-oriented programming language. However, as of Java 5.0, autoboxing enables programmers to proceed as if primitive types are instances of their wrapper classes.

**MySQL**

MySQL is a relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL is owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Sun Microsystems, a subsidiary of Oracle Corporation.

Members of the MySQL community have created several forks such as Drizzle and MariaDB. Both forks were in progress long before the Oracle acquisition (Drizzle was announced 8 months before the Sun acquisition).

Free-software projects that require a full-featured database management system often use MySQL. Such projects include (for example) WordPress, phpBB and other software built on the LAMP software stack. MySQL is also used in many high-profile, large-scale World Wide Web products including Wikipedia, Google, Drupal and Face book.

**Uses**

Many web applications use MySQL as the database component of a LAMP software stack. Its popularity for use with web applications is closely tied to the popularity of PHP, which is often combined with MySQL. Several high-traffic web sites (including Flickr, Facebook, Wikipedia, Google (though not for searches), Nokia and YouTube[11]) use MySQL for data storage and logging of user data.

**Platforms and interfaces**

MySQL code uses C and C++. The SQL parser uses yacc and a home-brewed lexer, sql\_lex.cc.

MySQL works on many different system platforms, including AIX, BSDi, FreeBSD, HP-UX, i5/OS, Linux, Mac OS X, NetBSD, Novell NetWare, OpenBSD, OpenSolaris, eComStation, OS/2 Warp, QNX, IRIX, Solaris, Symbian, SunOS, SCO OpenServer, SCO UnixWare, Sanos, Tru64 and Microsoft Windows. A port of MySQL to OpenVMS also exists.

All major programming languages with language-specific APIs include Libraries for accessing MySQL databases. In addition, an ODBC interface called MyODBC allows additional programming languages that support the ODBC interface to communicate with a MySQL database, such as ASP or ColdFusion. The MySQL server and official libraries are mostly implemented in ANSI C/ANSI C++.

**Management and Graphical Frontends**

MySQL Workbench in Windows, displaying the Home Screen which streamlines use of its full capabilities

MySQL is primarily an RDBMS and therefore ships with no GUI tools to administer MySQL databases or manage data contained within. Users may use the included command-line tools, or download MySQL Frontends from various parties that have developed desktop software and web applications to manage MySQL databases, build database structure, and work with data records.

**Official**

The official MySQL Workbench is a free integrated environment developed by MySQL AB, that enables users to graphically administer MySQL databases and visually design database structure. MySQL Workbench replaces the previous package of software, MySQL GUI Tools. Similar to other third-party packages but still considered the authoritative MySQL frontend, MySQL Workbench lets users manage the following:

* Database design & modeling
* SQL development — replacing MySQL Query Browser
* Database administration — replacing MySQL Administrator

MySQL Workbench is available in two editions, the regular free and open source Community Edition which may be downloaded from the MySQL website, and the proprietary Standard Edition which extends and improves the feature set of the Community Edition.

**Third party**

Several other third-party proprietary and free graphical administration applications (or "Frontends") are available that integrate with MySQL and enable users to work with database structure and data visually. Some well-known frontends are:

* phpMyAdmin - a free Web-based frontend widely installed by Web hosts worldwide, since it is developed in PHP and only requires the LAMP stack to run.
* HeidiSQL - a full featured free frontend that runs on Windows, and can connect to local or remote MySQL servers to manage databases, tables, column structure, and individual data records. Also supports specialised GUI features for date/time fields and enumerated multiple-value fields.
* Navicat - a series of proprietary graphical database management applications, developed for Windows, Macintosh and Linux.
* Other available proprietary MySQL frontends include Adminer, Aqua Data Studio, dbForge Studio for MySQL, Epictetus, Oracle SQL Developer, SchemaBank, SQLyog, SQLPro SQL Client, Toad and Toad Data Modeler.

**Deployment**

MySQL can be built and installed manually from source code, but this can be tedious so it is more commonly installed from a binary package unless special customizations are required. On most Linux distributions the package management system can download and install MySQL with minimal effort, though further configuration is often required to adjust security and optimization settings.

Though MySQL began as a low-end alternative to more powerful proprietary databases, it has gradually evolved to support higher-scale needs as well.

It is still most commonly used in small to medium scale single-server deployments, either as a component in a LAMP based web application or as a standalone database server. Much of MySQL's appeal originates in its relative simplicity and ease of use, which is enabled by an ecosystem of open source tools such as phpMyAdmin.

In the medium range, MySQL can be scaled by deploying it on more powerful hardware, such as a multi-processor server with gigabytes of memory.

There are however limits to how far performance can scale on a single server, so on larger scales, multi-server MySQL deployments are required to provide improved performance and reliability. A typical high-end configuration can include a powerful master database which handles data write operations and is replicated to multiple slaves that handle all read operations. The master server synchronizes continually with its slaves so in the event of failure a slave can be promoted to become the new master, minimizing downtime. Further improvements in performance can be achieved by caching the results from database queries in memory using memcached, or breaking down a database into smaller chunks called shards which can be spread across a number of distributed server clusters.

**Features**

As of April 2009, MySQL offers MySQL 5.1 in two different variants: the MySQL Community Server and Enterprise Server. They have a common code base and include the following features:

* A broad subset of ANSI SQL 99, as well as extensions
* Cross-platform support
* Stored procedures
* Triggers
* Cursors
* Updatable Views
* True Varchar support
* INFORMATION\_SCHEMA
* Strict mode
* X/Open XA distributed transaction processing (DTP) support; two phase commit as part of this, using Oracle's InnoDB engine
* Independent storage engines (MyISAM for read speed, InnoDB for transactions and referential integrity, MySQL Archive for storing historical data in little space)
* Transactions with the InnoDB, BDB and Cluster storage engines; savepoints with InnoDB
* SSL support
* Query caching
* Sub-SELECTs (i.e. nested SELECTs)
* Replication support (i.e. Master-Master Replication & Master-Slave Replication) with one master per slave, many slaves per master, no automatic support for multiple masters per slave.
* Full-text indexing (Index\_(database)) and searching using MyISAM engine
* Embedded database library
* Partial Unicode support (UTF-8 and UCS-2 encoded strings are limited to the BMP)
* Partial ACID compliance (full compliance only when using the non-default storage engines InnoDB, BDB and Cluster)
* Shared-nothing clustering through MySQL Cluster
* Hot backup (via mysqlhotcopy) under certain conditions

The developers release monthly versions of the MySQL Enterprise Server. The sources can be obtained either from MySQL's customer-only Enterprise site or from MySQL's Bazaar repository, both under the GPL license. The MySQL Community Server is published on an unspecified schedule under the GPL and contains all bug fixes that were shipped with the last MySQL Enterprise Server release. Binaries are no longer provided by MySQL for every release of the Community Server.

**Distinguishing features**

MySQL implements the following features, which some other RDBMS systems may not:

* Multiple storage engines, allowing one to choose the one that is most effective for each table in the application (in MySQL 5.0, storage engines must be compiled in; in MySQL 5.1, storage engines can be dynamically loaded at run time):
  + Native storage engines (MyISAM, Falcon, Merge, Memory (heap), Federated, Archive, CSV, Blackhole, Cluster, Berkeley DB, EXAMPLE, and Maria)
  + Partner-developed storage engines (InnoDB, solidDB, NitroEDB, Infobright (formerly Brighthouse), Kickfire, XtraDB, IBM DB2[22])
  + Community-developed storage engines (memcache\_engine, httpd, PBXT, Revision Engine)
  + Custom storage engines
* Commit grouping, gathering multiple transactions from multiple connections together to increase the number of commits per second.

**Product History**

Milestones in MySQL development include:

* Original development of MySQL by Michael Widenius and David Axmark beginning in 1994
* First internal release on 23 May 1995
* Windows version was released on 8 January 1998 for Windows 95 and NT
* Version 3.23: beta from June 2000, production release January 2001
* Version 4.0: beta from August 2002, production release March 2003 (unions)
* Version 4.01: beta from August 2003, Jyoti adopts MySQL for database tracking
* Version 4.1: beta from June 2004, production release October 2004 (R-trees and B-trees, subqueries, prepared statements)
* Version 5.0: beta from March 2005, production release October 2005 (cursors, stored procedures, triggers, views, XA transactions)

The developer of the Federated Storage Engine states that "The Federated Storage Engine is a proof-of-concept storage engine", but the main distributions of MySQL version 5.0 included it and turned it on by default. Documentation of some of the short-comings appears in "MySQL Federated Tables: The Missing Manual".

* Sun Microsystems acquired MySQL AB on 26 February 2008.
* Version 5.1: production release 27 November 2008 (event scheduler, partitioning, plugin API, row-based replication, server log tables)

Version 5.1 contained 20 known crashing and wrong result bugs in addition to the 35 present in version 5.0.

MySQL 5.1 and 6.0 showed poor performance when used for data warehousing — partly due to its inability to utilize multiple CPU cores for processing a single query.

* Oracle acquired Sun Microsystems on January 27, 2010.Oracle and Sun

**Future releases**

The MySQL 6 roadmap outlines support for:

* Referential integrity and Foreign key support for all storage engines is targeted for release in MySQL 6.1 (although it has been present since version 3.23.44 for InnoDB).
* Support for supplementary Unicode characters, beyond the 65,536 characters of the Basic Multilingual Plane (BMP); announced for MySQL 6.0.
* A new storage engine called Falcon. A preview of Falcon is available on MySQL's website.

The 2006 roadmap for future versions plans support for parallelization.

**Support and licensing**

Via MySQL Enterprise MySQL AB offers support itself, including a 24/7 service with 30-minute response time. The support team has direct access to the developers as necessary to handle problems. In addition, it hosts forums and mailing lists, employees and other users are often available in several IRC channels providing assistance.

In addition to official product support from Sun, other companies offer support and services related to usage of MySQL. For example, Pythian offers full database administration, architecture, optimization and training services. Percona and 42sql offer services related to optimization and Monty Program Ab offers non-recurring engineering such as patches to MySQL. OpenQuery provides MySQL training.

Buyers of MySQL Enterprise have access to binaries and software certified for their particular operating system, and access to monthly binary updates with the latest bug-fixes. Several levels of Enterprise membership are available, with varying response times and features ranging from how to and emergency support through server performance tuning and system architecture advice. The MySQL Network Monitoring and Advisory Service monitoring tool for database servers is available only to MySQL Enterprise customers.

Potential users can install MySQL Server as free software under the GNU General Public License (GPL), and the MySQL Enterprise subscriptions include a GPL version of the server, with a traditional proprietary version available on request at no additional cost for cases where the intended use is incompatible with the GPL.

Both the MySQL server software itself and the client libraries use dual-licensing distribution. Users may choose the GPL,[29] which MySQL has extended with a FLOSS License Exception. It allows Software licensed under other OSI-compliant open source licenses, which are not compatible to the GPL, to link against the MySQL client libraries.

Customers that do not wish to follow the terms of the GPL may purchase a proprietary license.Like many open-source programs, MySQL has trademarked its name, which others may use only with the trademark holder's permission.

**SOFTWARE TESTING**

**GENERAL**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

**DEVELOPING METHODOLOGIES**

The test process is initiated by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used.

The process verifies that the application meets the requirements specified in the system requirements document and is bug free. The following are the considerations used to develop the framework from developing the testing methodologies.

**Types of Tests**

**Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produces valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**Functional test**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

**System Test**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**Performance Test**

The Performance test ensures that the output is produced within the time limits, and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results.

**Integration Testing**

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

***Acceptance Testing***

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Acceptance testing for Data Synchronization:**

* The Acknowledge will be received by the Sender Node after the Packets are received by the Destination Node
* The Route add operation is done only when there is a Route request in need
* The Status of Nodes information is done automatically in the Cache Updating process

**Build the test plan**

Any project can be divided into units that can be further performed for detailed processing. Then a testing strategy for each of this unit is carried out. Unit testing helps to identity the possible bugs in the individual component, so the component that has bugs can be identified and can be rectified from errors.

**FUTURE WORK:**

Time Series Forecasting is one of the least explored areas and various models are evaluated to improve the accuracy of the forecast. The main focus of the proposal is to predict future web traffic to make decisions for better congestion control. Past Values are considered to predict future values. We will also seek to explore multivariate time series and offer suggestions for simplifying the decision-making process in real-time.

**CONCLUSION:**

Web traffic Time arrangement forecast can be completed utilizing Long Short Term Memory Recurrent Neural Network furthermore, Autoregressive coordinated moving normal all the more productively and precisely. Expectation of the quantity of clients will get to the website later on is conceivable. The proposed will continue improving as more client information is taken care of. Our framework can be utilized across all websites for improving their web traffic load the executives and business analysis[5]. LSTM RNN carries more proficiency to our framework. Our framework successfully catches occasional examples furthermore, long haul patterns Including data about occasions, day of week, language, locale may help our model to catch all the more effectively the highs and lows.

**References:**

[1] "Predicting Computer Network Traffic: A Time Series Forecasting Approach using DWT, ARIMA and RNN" by Rishabh Madan, 2018.

[2] “Fast ES-RNN: A GPU Implementation of the ES-RNN algorithm ” by Andrew Redd and Kaung Khin, 2019.

[3] “Time Series Forecasting Based on Complex Network Analysis“ by SHENGZHONG MAO AND FUYUAN XIAO, 2019.

[4] “Web Traffic Prediction of Wikipedia Pages” by Navyasree Petluri, Eyhab Al-Masri, 2019.

[5] “Time series forecasting using improved ARIMA” by Soheila Mehrmolaei,2016.

[6] “Efficient Prediction ofy Network Traffic for Real-Time Applications” by Muhammad Faisal Iqbal , Muhammad Zahid, Durdana Habib, and Lizy Kurian John, 2019.

[7] https://wikitech.wikimedia.org/wiki/Analytics/AQS/Pageviews

[8] https://towardsdatascience.com/3-facts-about-timeseries-forecasting-that-surprise-experienced-machinelearning-practitioners-69c18ee89387.

[9] "Temporal Pattern Attention for Multivariate Time Series Forecasting" by Shun-Yao Shih Fan-Keng Sun Hung-yi Lee, 2018.

[10]“Time series forecasting using improved ARIMA” by Soheila Mehrmolaei,2016.

[11] “Efficient Prediction of Network Traffic for Real-Time Applications” by Muhammad Faisal Iqbal , Muhammad Zahid, Durdana Habib, and Lizy Kurian John, 2019.

[12] “Modelling Approaches for Time Series Forecasting and Anomaly Detection” by Shuyang Du , Madhulima Pandey, and Cuiqun Xing, 2018.

[13] “Neural Decomposition of Time-Series Data for Effective Generalization” by Luke B. Godfrey and Michael S. Gashler, 2017