Real-Time Vehicle Traffic Analysis using Long Short Term Memory Networks in Apache Spark

Anveshrithaa S
School of Computer Science & Engineering
Vellore Institute of Technology
Vellore, India
anveshrithaas@gmail.com

Lavanya K
School of Computer Science & Engineering
Vellore Institute of Technology
Vellore, India
lavanya.k@vit.ac.in

2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE) 978-1-7281-4142-8/20/$31.00 ©2020 IEEE 10.1109/ic-ETITE47903.2020.97

***Abstract*—Escalating traffic congestion in large and rapidly evolving metropolitan areas all around the world is one of the inescapable problems in our daily lives. In light of this situation, traffic monitoring and analytics is becoming the need of the hour. Real-time traffic analysis requires the processing of data streams that are generated continuously to gain quick insights. In order to processes stream data at a faster rate, we need technologies with high computing capacity. Big data frameworks such as Apache Hadoop, Spark and Kafka, with their capability of processing a massive amount of data, have made it possible to develop advanced and efficient data stream processing systems. The challenge of analyzing data streams for real-time prediction can be overcome by exploiting deep learning techniques. Taking this as a motivation, this work aims at developing a real-time data stream processing model for forecasting vehicle traffic, using Long Short-Term Memory (LSTM) networks to learn and train itself from traffic data. In the proposed analytical framework, the traffic data from an API is streamed using a distributed streaming platform called Kafka into the machine learning model in Apache Spark for analysis. The proposed model is aimed at predicting traffic flow information by integrating Spark and Kafka along with deep neural networks, that will be of great value to the citizens as well as the government by reducing the travel time, cost and energy, thus having a positive impact on the environment and the society.**

Keywords—Apache Kafka, LSTM, neural networks, real-time analytics, Spark streaming, traffic prediction

I. INTRODUCTION

With the dawn of the era of Big Data upon us, there has been a vast increase in the amount of available data and it is anticipated to rise several folds in the approaching years. Extensive research is being carried out to put this data into good use. The growth in big data and big data techniques have opened up interesting opportunities that were once unimagined by researchers and businesses. Research on how to make use of the available data with Artificial Intelligence is achieving fascinating and noteworthy results. One of the areas which needs the attention of this is vehicular traffic analysis which has emerging demands in today's world. According to the World Urbanization Prospects by United Nations, the population of the world living in urban areas has surged swiftly from 751 million in 1950 to 4.2 billion in 2018 which constitutes 55% of the world's population, a proportion that is expected to rise to 68% by 2050 [1]. With such rapid urbanization, vehicular traffic on roads is set to increase drastically leading to congestion and accidents. This issue of increasing vehicle traffic needs to be addressed by exploiting machine learning and big data technologies to offer efficient solutions that provide convenience and efficiency for commuters. With the easy and wide accessibility of huge chunks of data along with the progress in big data analytics, it has been made possible to bring advancements in traffic analysis. Though the advancement of technology has already started creating a great impact on the field of transportation and traffic management, by addressing problems and providing effective, advanced and intelligent solutions, it is still a demanding problem to carry out real-time traffic analysis. This is because of the intricacies in analyzing and learning from huge volumes of real-time streaming data and providing accurate predictions which needs more consideration. This can be expedited by leveraging Machine Learning and big data techniques like stream processing to develop better solutions. With the advances in various big data frameworks that process large quantities of data in a distributed computing environment, it has become easy to manage, process, analyze and store data for real-time analytics. These big data technologies along with the tremendous growth in the field of artificial intelligence, have made it possible to develop efficient stream processing systems that can curb the problem of increasing vehicular traffic congestion around the world by real-time traffic analysis.

This work focuses on traffic analysis by utilizing a Machine Learning technique and an efficient cluster computing, big data framework based on Hadoop MapReduce called Apache Spark that effectively handles unstructured, real-time, streaming data.

A. Apache Spark Ecosystem

Spark incorporates various in-built libraries and components that include the Spark core, Spark Streaming, Spark MLLib, Spark SQL, and Spark GraphX. Fig. 1 represents the Spark ecosystem.

i) Spark Core:

The basic functionalities of Spark are built upon the Spark Core. It incorporates Resilient Distributed Datasets (RDDs) which is one of Spark's main abstractions. It is responsible for task scheduling, fault recovery, in-memory computation, and memory management. It is the foundation for the processing of large datasets.

ii) Spark SQL:

It is a module of Spark that provides SQL interface to Spark for working with structured and semi-structured data and executing SQL queries on them.

978-1-7281-4142-8/$31.00 ©2020 IEEE

iii) Spark Streaming:

It is a Spark component built upon the Spark core that is responsible for high- throughput, scalable and fault-tolerant stream processing of continuously flowing data streams obtained from data streaming sources like Apache Kafka, Apache Flume and Amazon Kinesis.

iv) Spark MLlib:

It is a Spark library that facilitates the implementation of Machine Learning algorithms and makes Machine Learning scalable. This package of Spark is Dataframe-based rather than using RDD.

v) Spark GraphX:

It is a Spark API for manipulating and working with graphs and executing graph-parallel computations.

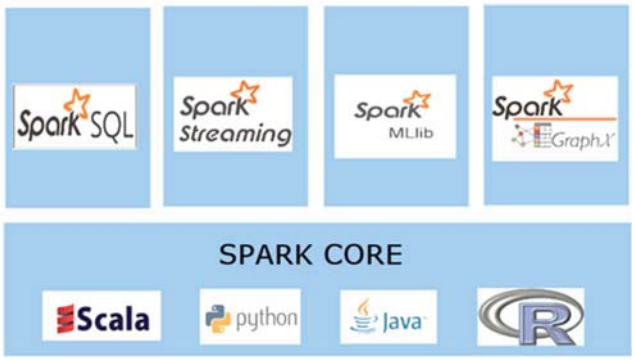

Fig. 1. Apache Spark ecosystem

In the case of real-time traffic analysis, new and dynamic data is continuously being generated by IoT devices and sent to a data streaming application such as a third-party API from where Spark streaming receives continuously flowing data for processing and performing real-time analytics. The data from the API can be streamed into the spark application using a publish-subscribe messaging platform called Apache Kafka which is a fast, scalable and fault-tolerant streaming platform that uses real-time data pipelines with low latency and high throughput that can be integrated with distributed stream processing frameworks such as Spark for real-time ingesting and processing of data streams.

The proposed work is implemented by integrating Apache Spark, Apache Kafka and MongoDB along with LSTM networks for efficiently performing real-time traffic analysis and forecast and then storing the data in a database. This system is developed using Spark MLlib to process and analyze the stream data sequentially and incrementally and Spark SQL for executing SQL queries on them.

II. Literature Survey

First, Prathilothamai et al., presented a cost-effective model for the prediction the traffic condition on roads using Apache Spark. The data is collected using sensors and the collected sensor data is then processed and converted into a CSV file which is processed in Apache Spark for prediction of traffic volume as low, medium and high [2]. Paulo et al., presented a scalable system architecture reinforced by Big Data tools and techniques, that is capable of processing real-time traffic data. This system performs real-time processing of traffic-related streaming data and real-time traffic prediction, using stream mining. Results obtained from this work suggest that it is advantageous to use data stream management system, compared to traditional database management systems [3]. Guerreiro et al., proposed an architecture that uses big data frameworks, to analyse and store massive amounts of traffic-related data from various data sources. It presents a stream processing approach for an intelligent transportation system. The proposed architecture has the ability to handle real-time data along with historical traffic data by means of big data tools like Apache Spark and MongoDB [4]. Mishra et al., proposed a work that uses a Complex Event Processing (CEP) engine that facilitates the interpretation of new states from arriving traffic data. This process of converting historical data to useful information is carried out by a LSTM network to predict the occurrence of an event in advance. The results of this work showcase the abilities of Deep Learning in predicting events ahead in time with minimal error [5].

Li et al., implemented an deep learning model that is based on parameter updating in order to facilitate real-time processing of data stream. The proposed system learns the features of dynamic data quickly and satisfies the instantaneous necessities of feature learning in big data at the same time retains the initial knowledge of the neural network. The performance of the system was tested on MNIST image data sets. The model is capable of learning the features of new data incrementally and it also has the capacity to learn and retain the original features of the data, increase the efficiency of the model, and maximize the performance in processing real-time data streams [6]. Amini et al., suggested a comprehensive and flexible model for big data analytics for traffic control in real-time with the use of distributed computing platform. This architecture is based on the methodical analysis of the requirements of the traffic control systems that are already existing. The architecture of the proposed system has been realized in a prototype platform that uses Kafka for stream processing [7]. Aydin et al., discussed a distributed system that runs on a cluster for collection, storage and analysis of sensor data using Apache Spark for processing and machine learning algorithms for analysis [8]. Xia et al., proposed a system in a MapReduce framework of on a Hadoop platform, in order to improve the efficiency and accuracy of short-term traffic flow prediction [9]. Biem et al., discussed a stream processing model that has the capacity to process a large amount of sensor data in real-time for traffic monitoring and planning based on IBM's System S platform [10]. Rathore et al., proposed a real-time stream processing technique by integrating the parallel and distributed environment of Spark with the GPU to improve the model and make it more efficient to handle a huge amount of high-speed data streaming [11]. Maarala et al., used Apache Spark for distributed and parallel computing, query, storage, ingestion, and processing real-time data streams. They proposed a real-time traffic data processing model in future work [12].

Nasiri et al., studied the use of distributed stream processing frameworks such as Apache Spark Streaming, Apache Storm, and Apache Flink for IoT applications in smart cities and evaluate their performances [13]. Parin and Pandi, presented a big data solution for road traffic monitoring using Hadoop technologies like Apache Map Reduce and Hive along with machine learning algorithms [14]. Zhou et al., proposed a network traffic monitoring system that uses stream processing frameworks like Spark Streaming for network analysis, monitoring and measurement [15]. Lee and Paik built a real time stock market analysis system to analyse

streaming twitter data in order to find the correlation with stock market analysis using big data techniques like Apache Spark [16]. Ali and Abdullah analysed and evaluated big data streaming analysis platforms used for real time analysis of data and presented solutions for online stream processing problems [17]. Lou et al., proposed an aquaculture monitoring system using Apache Flink, MongoDB, and Apache Kafka and then compared the efficiency of the proposed system with Hbase and concluded that their solution was more efficient for data storage and processing of aquaculture [18]. Yadranjiaghdam et al., proposed an analytical framework that ingests data using Apache Kafka, performs stream processing and machine learning algorithms on streaming Twitter data using Apache Spark for real-time analysis of tweets [19]. Jiskani et al., proposed an analytic framework in which machine learning method is integrated with Spark-based models, namely LSTM network and MLP to enhance the accuracy of prediction [20]. Zhao et al., presented a framework for real-time anomaly detection in network traffic using machine learning techniques. The proposed framework exploits big data technologies such as Apache Hadoop, Apache Kafka, and Apache Storm combined with machine learning algorithms for real-time analysis and processing of the real-time network-flow data [21]. Pekka conducted a feasibility analysis of big data tools and frameworks for processing streams of semi-structured data. AsterixDB, a big data management system for processing semi-structured data was compared with Spark streaming, that is integrated with a NoSQL database like Cassandra for stream-based processing [22].

III. Methodology

The proposed framework endeavors to analyse and predict vehicle traffic from streams of data from a third-party API that is streamed using Apache Kafka into the analytics model implemented in Spark using Long Short-Term Memory Networks (LSTM) and then store the original as well as the predicted data in a NoSQL database like MongoDB. Fig. 2 represents the architecture of the proposed system.

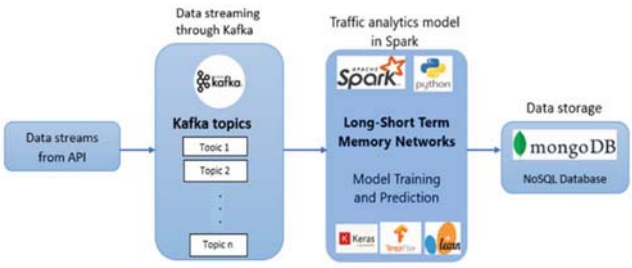

Fig. 2. Proposed system architecture

A. Data Streaming

The data analytics model uses Apache Kafka to continuously stream data from the source. Apache Kafka is a popular distributed streaming platform which adopts the 'publish-subscribe' messaging model to deal with the real-time volume of data. Kafka provides an API that facilitates applications to consume log events in real-time. Kafka maintains message categories called topics to which the producers publish data and the consumers subscribe to, in order to consume data. Kafka uses Zookeeper for coordination between the Kafka brokers and the consumers.

Firstly, the zookeeper instance is started and then the python producer script is executed to stream the traffic data from the API and publish it to the topic. The traffic data from the API contains various parameters such as lane type, free-flow speed, current travel time, free-flow travel time, etc. These continuous streams of messages are consumed from the topic subscribed by the Kafka consumer that is executed using a consumer script in python. The data is consumed by the spark streaming module that uses these input data streams to analyse and predict traffic flow.

B. Analytic model using LSTM in Spark

The Spark streaming consumes the data streams from the subscribed Kafka topics and the data is processed by the analytics model in Spark that uses Long Short-Term Memory Networks (LSTM) to predict the traffic flow from the input data streams. Long Short-Term Memory networks are a special type of Recurrent Neural Networks (RNNs) that overcome the vanishing gradients problem in RNNs. In LSTM, the flow of information is through gated cell states that allow it to selectively forget or remember things. The memory blocks in LSTM are called the cells and the mechanism responsible for the manipulations to these memory blocks is called gates. The three major gates are the input, output and the forget gate.

C. Data Storage

The processed data along with the original data from the data source is then stored in MongoDB which is a NoSQL database which can store large volumes of unstructured data in JSON format. The original data from the API streamed through Kafka is stored in the database with MongoDB as the sink, using a Kafka-MongoDB sink connector. Whereas, the forecasted data is stored into the data storage system from the Spark where it is being processed.

IV. Implementation and Results

The LSTM model uses 2 hidden layers with a dropout rate of 0.2 and sigmoid activation function. The RMSProp optimizer that uses a gradient-based optimization technique is used to train the model with 0.001 as the learning rate and a batch size of 256, over 600 epochs.

In order to train the model, we use the PeMs 5 minutes interval traffic dataset that contains data collected in real-time from the major metropolitan areas across the state of California. This training dataset contains traffic information like the lane type, average speed, average occupancy, lane flow, etc. for every 5 minutes for a period of around 60 days. After training, the model is then used on the data streams from the API for real-time analysis to determine the vehicle traffic on the roads.

A. Performance Metrics

The metrics considered to evaluate the model's performance are as follows:

1. Mean Absolute Error (MAE) is the mean of the absolute differences between the actual and the predicted observation given by

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - y'_i|$$

Where n is the number of observations, y_i is the i^{th} actual observation and y'_i is the i^{th} predicted value.

2. Root Mean Squared Error (RMSE) is the square root of the mean of the square of the differences between the actual and predicted values, given by the formula,

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - y'_i)^2}$$

3. Mean Absolute Percentage Error (MAPE) is a prediction accuracy metric used to measure the error in prediction as a percentage, whose formula is given by,

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - y'_i|}{y_i} * 100$$

4. R2 score
5. Explained variance score

After training the model on the training dataset for 600 iterations with a batch size of 256, it is tested on the test data to predict the traffic flow at various times of the day and the results are obtained as summarised in Table 1.

TABLE 1. PERFORMANCE METRICS OF THE MODEL

Metric	Value
MAE	7.5213
RMSE	10.3374
MAPE	17.2237%
R2 score	0.9342
Explained variance score	0.9344

Figures 3- 6 represent the plot of actual measured traffic flow against the predicted flow of traffic on a specefic day for a duration of every 6 hours of the day. The x-axis denotes the time of the day and the y-axis denotes the traffic flow in terms of the number of vehicles on the lane. Figure 7 shows the plot of the actual vs predicted traffic flow for a 24 hours duration on a particular day.

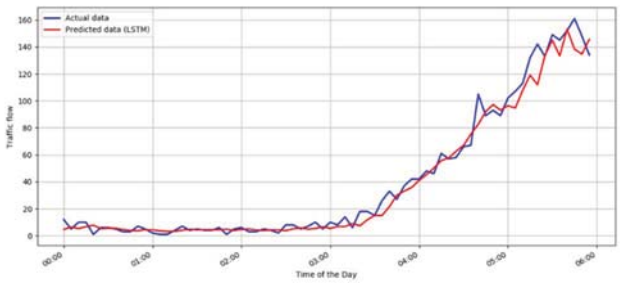

Fig. 3. Actual vs predicted traffic flow between 00:00 and 06:00 of a particular day

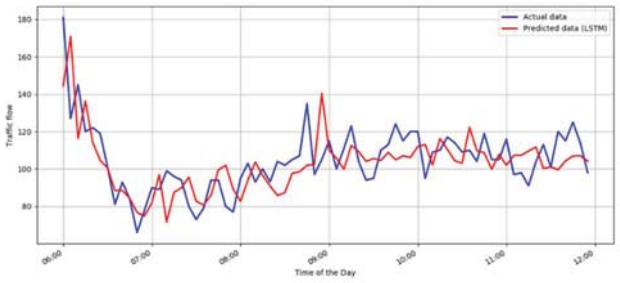

Fig. 4. Actual vs predicted traffic flow between 06:00 and 12:00 of a particular day

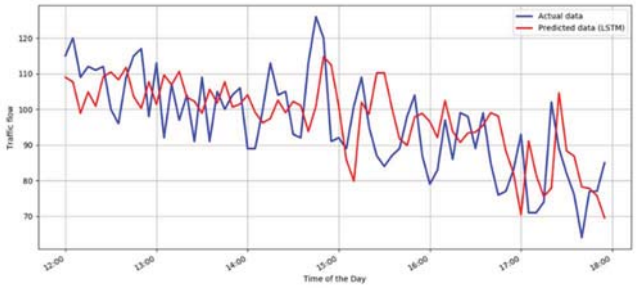

Fig. 5. Actual vs predicted traffic flow between 12:00 and 18:00 of a particular day

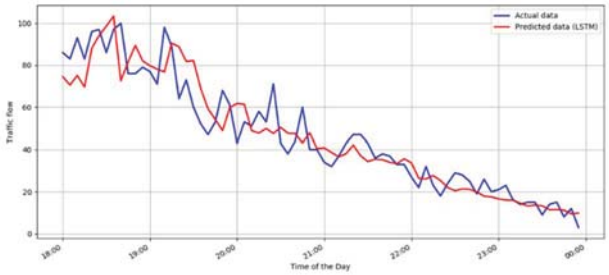

Fig. 6. Actual vs predicted traffic flow between 18:00 and 00:00 of a particular day

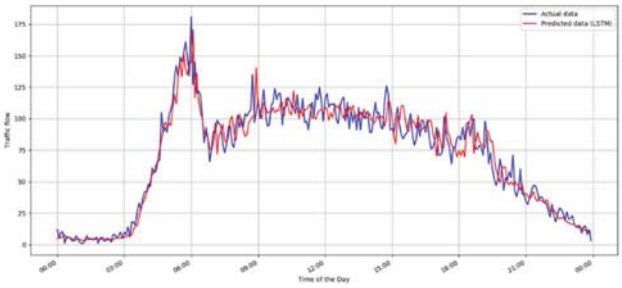

Fig. 7. Actual vs predicted traffic flow for a duration of 24 hours on a particular day.

V. CONCLUSION

Using Apache Spark, a traffic forecast model that leverages a deep learning model called Long- Short Term Memory network for predicting the flow of traffic on roads in real-time is implemented. With the use of Apache Kafka and Spark streaming, real-time predictive analysis of the traffic data was carried out and good performance of the model was observed in analyzing the data and predicting the flow of

vehicular traffic. To further improve the model performance, hyperparameter optimization can be carried out, to tune various parameters of the neural network like the number of layers, dropout rates, batch size, number of epochs, etc. for optimal performance in prediction of traffic data.

REFERENCES

[1] World Urbanization Prospectus, United Nations, 2018.

[2] M. Prathilothamai, A. M. Sree Lakshmi and Dilna Viswanthan, "Cost Effective Road Traffic Prediction Model using Apache Spark", Indian Journal of Science and Technology, Vol 9(17), 2016.

[3] P Figueiras, G Guerreiro, R Costa, Z Herga, A Rosa and R J Goncalves, "Real-Time Monitoring of Road Traffic using Data Stream Mining", IEEE International Conference on Engineering, Technology and Innovation, 2018.

[4] G Guerreiro, P Figueiras, R Silva, R Costa, R J Goncalves, "An architecture for big data processing on intelligent transportation systems. An application scenario on highway traffic flows", 2016 IEEE 8th International Conference on Intelligent Systems (IS), 2016.

[5] S Mishra, M Jain, B. S. N. Sasank, C. Hota, "An Ingestion Based Analytics Framework for Complex Event Processing Engine in Internet of Things", International Conference on Big Data Analytics, pp 266-281, 2018

[6] Y. Li, M. Zhang and W. Wang, "Online Real-Time Analysis of Data Streams Based on an Incremental High-Order Deep Learning Model," in IEEE Access, vol. 6, pp. 77615-77623, 2018.

[7] S. Amini, I. Gerostathopoulos and C. Prehofer, "Big data analytics architecture for real-time traffic control," 2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), Naples, 2017, pp. 710-715.

[8] G. Aydin, I. R. Hallac, and B. Karakus, "Architecture and Implementation of a Scalable Sensor Data Storage and Analysis System Using Cloud Computing and Big Data Technologies," Journal of Sensors, vol. 2015, Article ID 834217, 11 pages, 2015.

[9] D. Xia, B. Wang, H. Li, Y. Li, and Z. Zhang, "A distributed spatial-temporal weighted model on MapReduce for short-term traffic flow forecasting," Neurocomputing, vol. 179, pp. 246–263, 2016.

[10] Biem, E. Bouillet, H. Feng, A. Ranganathan, A. Riabov and O. Verscheure, "Real-Time Traffic Information Management using Stream Computing," IEEE Data Engineering Bulletin, vol. 33, no. 2, pp. 64-68, 2010.

[11] M. M. Rathore, H Son, A. Ahmad, A. Paul, G Jeon, "Real-Time Big Data Stream Processing Using GPU with Spark Over Hadoop Ecosystem", International Journal of Parallel Programming, vol. 46, Issue 3, pp 630–646, 2018.

[12] A. I. Maarala, M. Rautiainen, M. Salmi, S. Pirttikangas and J. Riekki, "Low latency analytics for streaming traffic data with Apache Spark," 2015 IEEE International Conference on Big Data (Big Data), Santa Clara, CA, 2015, pp. 2855-2858.

[13] H. Nasiri , S. Nasehi and M. Goudarzi, "Evaluation of distributed stream processing frameworks for IoT applications in Smart Cities", Journal of Big Data, 2019.

[14] Patel Parin and Gayatri Pandi, "Traffic Monitoring using Video Stream with Machine Learning: Based on Big Data Process with Cloud", International Journal of Innovations & Advancement in Computer Science (IJIACS), vol. 6, Issue 11, 2017.

[15] B. Zhou, J. Li, X. Wang, Y. Gu, L. Xu, Y. Hu, L. Zhu, "Online Internet traffic monitoring system using spark streaming," in Big Data Mining and Analytics, vol. 1, no. 1, pp. 47-56, March 2018.

[16] C. Lee and I. Paik, "Stock market analysis from Twitter and news based on streaming big data infrastructure," 2017 IEEE 8th International Conference on Awareness Science and Technology (iCAST), 2017, pp. 312-317.

[17] A. H. Ali and M. Z. Abdullah, "Recent Trends in Distributed Online Stream Processing Platform for Big Data: Survey," 2018 1st Annual International Conference on Information and Sciences,2018, pp. 140-145.

[18] Lou Y., Chen L., Ye F., Chen Y., Liu Z, "Research and Implementation of an Aquaculture Monitoring System Based on Flink, MongoDB and Kafka", International Conference on Computational Science, 2019, pp 648-657.

[19] B. Yadranjiaghdam, S. Yasrobi and N. Tabrizi, "Developing a Real-Time Data Analytics Framework for Twitter Streaming Data", IEEE International Congress on Big Data (BigData Congress), 2017, pp. 329-336.

[20] M.A. Jiskani, Karim M.R., Kim Y., 2018. "A Two-Stage Big Data Analytics Framework with Real World Applications Using Spark Machine Learning and Long Short-Term Memory Network", Symmetry,10(10), p. 485.

[21] S. Zhao, M. Chandrashekar, Y. Lee and D. Medhi, "Real-time network anomaly detection system using machine learning," 2015 11th International Conference on the Design of Reliable Communication Networks (DRCN), 2015, pp. 267-270.

[22] Pekka Pääkkönen, "Feasibility analysis of AsterixDB and Spark streaming with Cassandra for stream□based processing", Journal of Big Data, 2016, 3:6.