The American Journal of Engineering and Technology
ISSN 2689-0984 | Open Access

TYPE Original Research
PAGE NO. 105-111
DOI 10.37547/tajet/Volume07Issue03-09

Check for updates

Predicting Cargo Arrival Time Using Scala and Spark: Approaches and Achievements

Danylo Liakhovetskyi

Middle Java Backend Engineer at AgileEngine Pensacola, FL, USA

OPEN ACCESS

SUBMITTED 21 January 2025
ACCEPTED 15 February 2024
PUBLISHED 12 March 2025
VOLUME Vol.07 Issue03 2025

CITATION

Danylo Liakhovetskyi. (2025). Predicting Cargo Arrival Time Using Scala and Spark: Approaches and Achievements. The American Journal of Engineering and Technology, 105–111. https://doi.org/10.37547/tajet/Volume07Issue03-09

Abstract: The article examines methods to predict cargo arrival times through Apache Spark and Scala. The necessity for such methods arises due to external factors such as unpredictable road conditions, weather phenomena, and specific logistical operations. Information processing employs methods such as regression, decision trees, and neural networks, which analyze data from sensors, GPS devices, and other sources to build forecasts that consider all factors directly or indirectly affecting calculation accuracy.

The methodology is based on studying the functionality of the Apache Spark platform integrated with the Scala programming language, enabling the processing of large datasets with high operational speed and solution scalability.

The use of Apache Spark combined with Scala accounts for streaming data, which improves prediction accuracy. This method optimizes logistics processes by reducing delays and allowing timely responses to changes in external conditions.

The information presented in the article will be useful for data processing professionals, logisticians, and developers.

Keywords: ETA, Scala, Apache Spark, logistics, machine learning, big data.

Introduction: The need to predict the time of cargo delivery to its final destination is driven by the fact that arrival times directly impact the quality of transportation operations and the costs associated with these processes.

Technological advancements have fundamentally transformed the process of estimating delivery times by

leveraging machine learning technologies. These technologies are capable of processing large datasets directly sourced from various inputs, distinguishing them from traditional methods. The use of Apache Spark and Scala enables high-speed computations, which are essential for timely data processing and subsequent decision-making.

The relevance of this topic is determined by the need to process data obtained from various sources (such as GPS or meteorological indicators). The implementation of Apache Spark or Scala allows for real-time responses to ongoing changes and facilitates informed decision-making.

This study aims to analyze the opportunities that arise for companies through the application of Apache Spark and Scala in logistics.

MATERIALS AND METHODS

Predicting the arrival time of transportation vehicles holds significant potential in logistics, as it facilitates route planning and estimating approximate arrival times. For instance, in the studies by Li N. and Liao V. C. C. [1], [5], methods of data analysis are examined, including factors such as vessel parameters, weather conditions, and transportation history. These works propose using information from previous journeys to calculate arrival times with greater accuracy.

The articles by Khotimah H. and Sahoo R. [2], [3] analyze existing methods for predicting delays in air transportation. The authors note that machine learning algorithms enable the prediction of potential delays, while the use of the Apache Spark platform accelerates the processing of large datasets.

The study by Pérez-Chacón R. [4] focuses on the potential application of time series forecasting methods using Apache Spark. The implementation of distributed systems is shown to allow for the efficient handling of large datasets, ultimately improving the quality of data analytics.

However, despite advancements in prediction accuracy, unresolved challenges remain. Many studies focus exclusively on improving the precision of predictions, without giving adequate attention to the usability of models or their adaptability to modern conditions. Regarding the use of the Apache Spark platform and Scala, researchers note that this combination allows for efficient data processing, ensuring the performance and scalability needed to address information analysis tasks.

The methodology of this study involves the use of the Apache Spark platform with the Scala programming language, as this combination improves data processing speed and effectively addresses challenges related to information analysis.

RESULTS AND DISCUSSION

Traditional approaches previously used for predicting cargo arrival times rely on statistical models, which often fail to account for factors influencing transportation. As a result, these methods cannot promptly reflect changes in road conditions, weather, or the impact of external factors.

Apache Spark combined with Scala provides tools for data processing, enabling faster responses to ongoing changes. This allows for efficient handling of incoming information. However, to generate accurate predictions, it is essential to use data from various sources. This approach enables the system to adapt to changes, thereby improving the adjustment of delivery plans to reflect current conditions [1, 4, 5].

The process of predicting cargo arrival times using Scala and Apache Spark is illustrated in Figure 1.

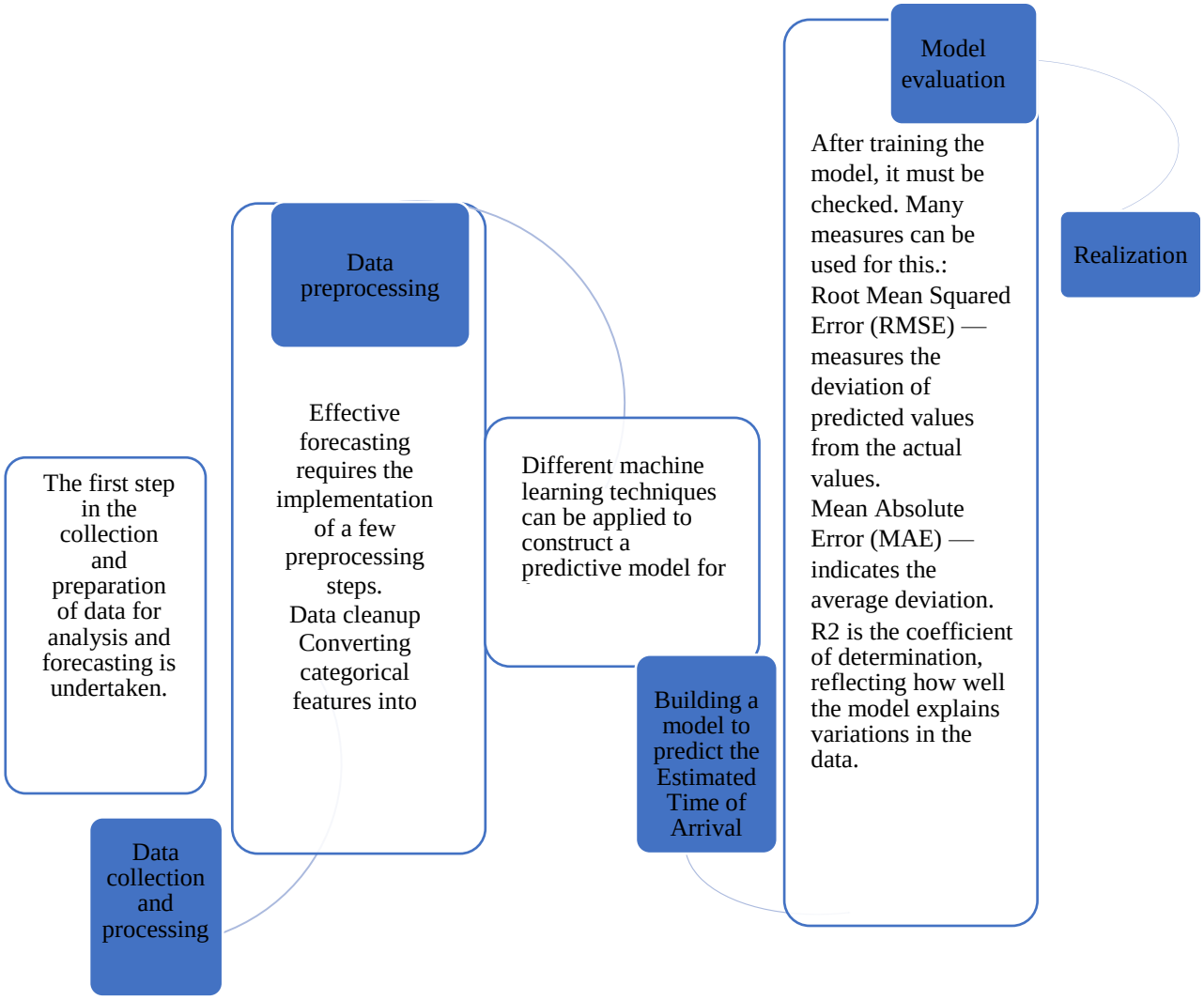

Fig. 1. The process of predicting the arrival time of goods using Scala and Apache Spark [1, 4, 5].

Apache Spark is a distributed computing platform designed for processing large volumes of data and creating predictive models in logistics. One of its key features is the ability to process streaming data through its Streaming component, enabling real-time handling of information such as transportation movement data, weather changes, road incidents, and other factors influencing logistical processes.

To generate arrival time predictions, Spark employs machine learning algorithms adapted for processing various types of information. The platform supports parallel processing, accelerating computations and enabling efficient handling of data under dynamic conditions. Spark integrates with multiple data sources, including transportation monitoring systems, GPS services, motion sensors, and Internet of Things devices. This integration facilitates the creation of models capable of responding to ongoing changes, thereby ensuring the accuracy of data analysis [2, 4, 5].

The Scala programming language is used for developing applications on the Apache Spark platform due to its capabilities for distributed computing and processing large data volumes. Scala supports both functional and object-oriented programming paradigms, making it suitable for solving tasks across various domains.

The integration of Scala with MLlib, the machine learning library in Spark, enhances the platform's ability to build predictive models using a variety of algorithms. Linear regression, random forests, gradient boosting, and neural networks are applied to create analytical solutions. The interaction between Spark and Scala forms the foundation for developing models that adapt

to data changes and respond to evolving conditions.

data processing, while its built-in tools for parallel computing enhance the efficiency of working with large datasets [2, 4, 5].

Scala is characterized by its concise syntax and high performance. Its functional capabilities optimize

The strengths and weaknesses of Scala and Apache Spark, as well as their impact on the process of predicting cargo arrival times, are detailed in Table 1.

Table 1. The strengths and weaknesses of Scala and Apache Spark, as well as the impact they have in predicting the arrival time of a shipment [1, 3, 5]

Parameter	Scala	Apache Spark
Strengths	- Functional and object-oriented programming styles	- High performance, achieved through distributed data processing
	- Scalability: capable of processing large datasets	- Fast data processing speed
	- Compatibility with Java and access to Java libraries	- Support for parallel computations and data processing
	- Multitasking and asynchronous operations via actors and Futures	- Optimized for handling large datasets
Weaknesses	- Steep learning curve for beginners	- Requires substantial resources for deployment
	- Less popular compared to Python or Java for machine learning	- High memory requirements for processing large datasets
	- Smaller ecosystem compared to other languages	- Dependency on cluster configurations and infrastructure for optimal performance
	- Limited support for visualization and subsequent data analysis tools	- Potential latency issues when processing small data volumes
Impact on predicting the arrival time of a shipment	- Enables the creation of high-performance algorithms optimized for data processing	- Capable of handling large datasets, necessary for making predictions considering factors that change in real-time
	- Facilitates integration with libraries for data analysis, simplifying the process of building predictive models	- Distributed computing allows training models on large datasets, increasing prediction accuracy
	- Suitable for developing flexible algorithms for data analysis and modeling	- Ideal for processing data from multiple sources, essential for creating forecasts that account for diverse factors

It is also worth noting that proprietary software was developed in Java for tracking systems in the logistics

sector to improve parcel tracking. For instance, machine learning algorithms were created to predict cargo arrival times. The application of machine learning methods for predicting cargo arrival times has become increasingly relevant due to the need to process large volumes of data. This field extensively employs both standard regression algorithms and advanced models, including ensembles, decision trees, and neural networks. These methods enable forecasts that account for various aspects of transportation processes.

Linear and polynomial regression serves as the foundation for predicting temporal metrics; however, such models are not always suitable for handling data with nonlinear dependencies. To address this limitation, other methods are employed to ensure greater flexibility in modeling.

Algorithms such as random forest and gradient boosting can identify hidden relationships within data, providing accurate results. These methods process multiple features, thereby improving the quality of predictions. Deep neural networks successfully handle multidimensional data, modeling complex dependencies. When these networks are combined with methods designed for time series analysis, they account for changes during transportation, which is critical for arrival time predictions. Adaptive methods are also employed to improve forecast accuracy, while recurrent neural networks, which account for temporal dependencies, enable more precise tracking of changes in the transportation process.

Once the data are prepared, the appropriate algorithm is selected to address the specific task. Regression methods are used for predicting delivery times or demand volumes, while classification algorithms are applied when object categorization is required, such as optimizing warehouse inventory storage. Clustering methods are employed for segmenting objects based on shared characteristics. For route optimization, algorithms such as genetic algorithms or simulated annealing are used.

After selecting the model, it is trained on prepared data and evaluated using various metrics. For regression tasks, the mean squared error is commonly applied, while for classification tasks, metrics such as accuracy, precision, recall, and the F1 score are used. The evaluation considers not only the model's performance on training data but also its ability to generalize to new, previously unseen data. Techniques such as cross-validation and A/B testing are employed for this purpose.

Upon completing the testing phase, the model is integrated into the company’s infrastructure, including ERP, WMS, and TMS systems. Integration via APIs allows the model to function by providing predictions that serve as the basis for decision-making. Post-deployment, the model’s performance is monitored, and its parameters are adjusted as needed [1, 4, 5].

The future trends in the use of Scala and Apache Spark will be presented in Table 2.

Table 2. Future Trends in the Use of Scala and Apache Spark (compiled by the author)

Name	Future Trends	Integration Features	Challenges	Minimization Approaches
Scala	- Enhanced support for machine learning and big data analytics	- Strong compatibility with Java, Python, and other programming languages for development and deployment	- Requires advanced qualifications for effective utilization of the programming language’s features	- Improved documentation and community expansion
	- Simplification and expansion of the ecosystem to support AI and ML	- Integration with Spark enables the use of its data processing capabilities	- Limited documentation for new features and tools in Scala	- Engaging experts and enhancing programming education programs

	- Integration with high-performance computing	- Support for data analytics libraries and frameworks such as Breeze and Akka	- Rapid development of new frameworks may cause library and tool obsolescence	- Development of new tools and libraries to enhance developer productivity
Apache Spark	- Advancements in integration with analytical tools	- Integration with various data storage systems and processing tools to create a unified data processing system	- Performance optimization issues when using Spark	- Development of optimized tools for small-scale data processing in Spark to reduce latency
	- Expansion of streaming data processing capabilities	- Ease of integration with other machine learning systems and analytics tools	- Challenges with configuration and scaling of distributed computations, particularly in large clusters	- Automation of scaling and configuration to improve the ease of handling large data volumes
	- Forecasting and subsequent data processing using AI and ML	- Interaction with libraries for analytics and visualization	- Compatibility with other platforms and data sources may complicate scaling	- Use of containerization to improve integration and deployment across various platforms

Thus, the potential of machine learning is evident in increasing efficiency, improving forecast accuracy, and enhancing service quality. Advancements in technology facilitate the application of complex models, ensuring adaptability in solving logistical challenges.

CONCLUSION

The study examined the prediction of cargo arrival times using the Apache Spark platform and the Scala programming language. An analysis of traditional methods revealed their limited applicability in the face of changing factors such as road conditions, weather fluctuations, and transportation flow instability. The application of data processing methods and machine learning technologies improves prediction accuracy and ensures model flexibility under changing external conditions.

Apache Spark serves as a platform for distributed data processing, enabling real-time handling of streaming information. Scala, integrated with the Spark ecosystem, offers capabilities for developing models that account for diverse relationships in large datasets. This combination enhances performance and facilitates achieving high-quality results in data analysis.

The results demonstrated that using the Spark platform and the Scala language supports the integration of data from various sources. Adaptive models, capable of responding to changes in external conditions such as weather fluctuations or traffic congestion, optimize logistical processes and enable more efficient resource utilization.

REFERENCES

Li N. et al. Modeling categorized truck arrivals at ports: Big data for traffic prediction //IEEE Transactions on Intelligent Transportation Systems. – 2022. – Vol. 24 (3). – pp. 2772-2788.

Khotimah H. et al. Performance Analysis of the Distributed Support Vector Machine Algorithm Using Spark for Predicting Flight Delays //E3S Web of Conferences. – EDP Sciences. - 2023. – Vol. 465, 02037. – pp.1-10

Sahoo R. et al. A hybrid ensemble learning-based prediction model to minimise delay in air cargo transport using bagging and stacking //International Journal of Production Research. – 2022. – Vol. 60 (2). – pp. 644-660.

Pérez-Chacón R. et al. Big data time series forecasting based on pattern sequence similarity and its application to the electricity demand //Information Sciences. – 2020. – Vol. 540. – pp. 160-174.

Liao V. C. C. Artificial Intelligence Technology to Predict Exact Estimated Time of Arrival for Smart Transportation Using Past Shipment Data //2024 IEEE 4th International Conference on Electronic Communications, Internet of Things and Big Data (ICEIB). – IEEE. - 2024. – pp. 275-277.