ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

UNIVERSITY OF PIRAEUS

Thesis

Implementation of the TRACLUS algorithm in Neural Networks for vessel trajectory prediction.

Lazaros S. Sofikitis

AM P12143

supervisors:

Eva Chondrodima

Yannis Theodoridis

athens, 2023

Abstract

This thesis explores the use of machine learning and clustering algorithms for predicting and improving vessel trajectories. Initially, a machine learning model is trained on vessel trajectory data to predict future vessel locations. Then the trajectories are clustered using a clustering algorithm called TRACLUS, which groups similar vessel trajectories together based on their location, velocity, and trajectory patterns. The thesis then evaluates the effectiveness of the clustering algorithm by comparing the predicted results of the model before and after the trajectories are clustered. The results show that the use of the TRACLUS algorithm improves the accuracy of the model, indicating that clustering can help identify patterns and relationships within vessel trajectories that are difficult to identify through machine learning models alone. Overall, this thesis demonstrates the potential of combining machine learning and clustering techniques for improving vessel trajectory predictions and enhancing our understanding of vessel behavior in marine transportation systems.

To my professors, supervisors and family
for their unwavering help and support.

Contents

Chapter 1

Introduction

Maritime transportation systems are necessary for human mobility. An important aspect of maritime transportation systems is the accurate prediction of ship routes. However, accurate ship route prediction poses challenges due to the complex and dynamic changes in marine traffic conditions. Machine learning methods can harness the massive ”data explosion” in vessel monitoring to enhance and facilitate the digitization of the shipping industry and address the problem of ship route prediction. This thesis aims to predict the position and/or route of ships using machine learning methods and by implementing a trajectory clustering algorithm.

1. Definition of the trajectory prediction problem.

Given a) a trajectory of a moving object: $\{p_0, t_0, p_1, t_0 + Dt_1, ..., p_i, t_{i-1} + Dt_i\}$, consisting of i transitions of the object, and b) a time interval Dt_{i+1}, the goal is to predict the expected position p_{i+1} of the object at the time $t_{i+1} = t_i + Dt_{i+1}$. Practically, given the ”when” component, the goal is to predict the corresponding ”where” component, e.g. where the vessel will be in the next few minutes.

Chapter 2

Machine learning and Neural Networks

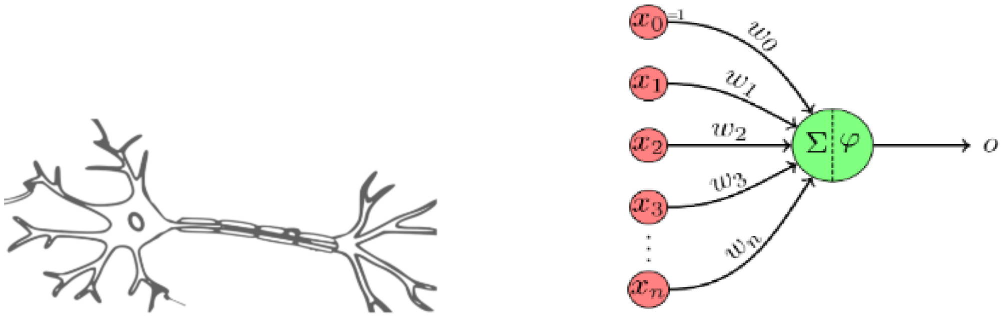
Machine learning is a branch of artificial intelligence (AI) and computer science that focuses on the use of data and algorithms to imitate the way that humans learn and through iteration, gradually improve its accuracy. One popular type of machine learning is neural networks. Neural networks are computing systems inspired by the neural networks that constitute biological brains. A Neural Network (NN) is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain. NNs are able to recognise patterns and data, make predictions and even generate new data.

1. Machine Learning

Machine learning applies algorithms and statistical models to interpret data and improve performance on a specific task. The main goal of machine learning is the development of algorithms that can make correct predictions on data without being specifically programmed to do so. This can provide deeper insights on problems and better understanding between causation and correlation of variables Fundamentally, machine learning focuses on building models that can generalise and adapt to examples in a dataset, with the goal of making accurate predictions and decisions about new data. The life cycle of machine learning models includes several steps: problem definition, data collection, data preprocessing, model training, model evaluation, model deployment, and model maintenance. In general, the creation and maintenance of a machine learning model is an iterative process with the focus on constant and continuous improvements to achieve optimal performance on the task at hand. There are several different types of machine learning algorithms; the most common types include reinforcement learning, unsupervised learning, and supervised learning. Reinforcement algorithms use a trial-and-error technique and aim to maximise a reward based on their actions. Unsupervised techniques learn from unlabeled data and aim to find patterns and structure in the data; on the other hand, supervised techniques use labelled examples where the input data is associated with the output target data. Machine learning has many practical applications in various domains, including healthcare, finance, marketing, natural language processing, and automotive. Most notably, it has been used for tasks such as image recognition, speech recognition, fraud detection, and driving autonomous cars. In this study, the focus will be on supervised training and its application to sequential problems. Sequential problems involve using an input sequence to output a prediction. An example of that could be predicting the future values of a time series based on past observations.

2. Neurons | Perceptrons

Neurons, also called perceptrons, are the building blocks of a neural network; in operation, they mimic the neurons of the brain. Within a neural network, neurons are organized into layers. The input layer receives input data, such as an image, number, or text, and passes it to the first hidden layer of neurons. Each subsequent layer processes the output of the previous layer and passes it to the next layer, until the final output layer produces a prediction or classification. Specifically, the neuron is a set of inputs, a set of weights, and an activation function. These inputs can either be raw input features or the output of neurons from an earlier layer. The neuron translates these inputs into a single output, which is then passed through an activation function and is picked up as input for another layer of neurons. Each neuron has a weight vector for every input to that neuron. The weights and biases for each neuron are adjusted based on the error between the predicted and actual output during the training stage such that the final network output is biased toward some preferred value.



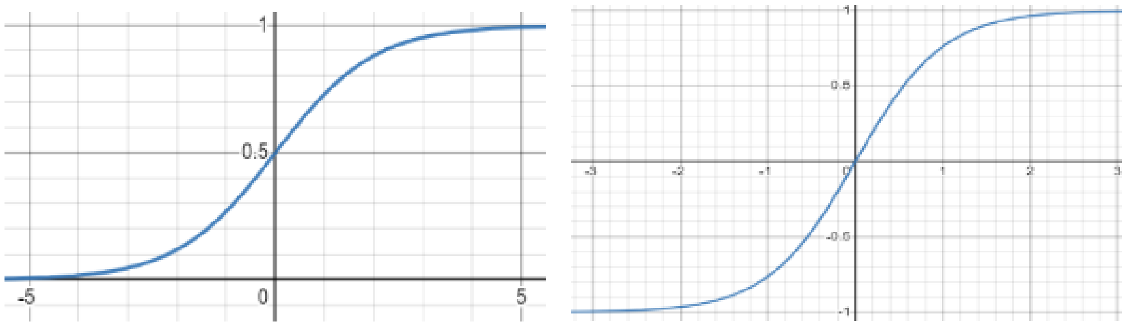
(a) The structure of a neuron cell

(b) The structure of a perceptron

Figure 2.1: The similarities between biological neurons and perceptrons.

3. Activation Functions

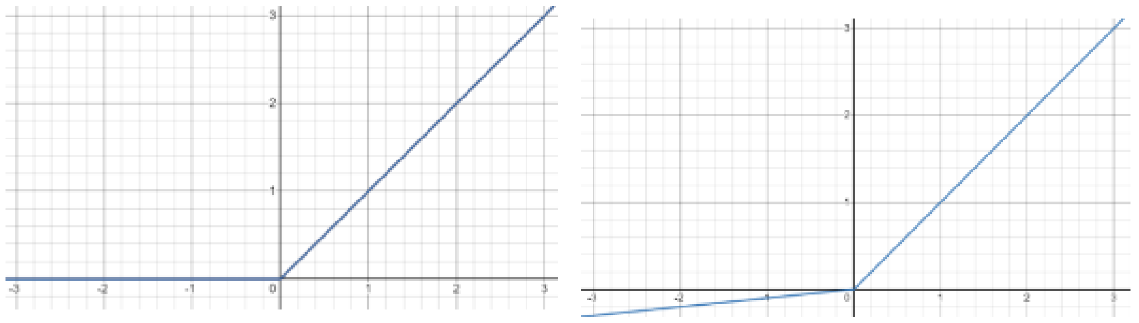
The output of each neuron passes through an activation function before continuing to the next layer of neurons. The purpose of an activation function is to introduce nonlinearity into the output of a neuron, allowing the network to model complex relationships between inputs and outputs. Without an activation function, a neural network would simply be a linear regression model. Typically, activation functions are nonlinear and monotonically increasing . Some commonly used activation functions include the sigmoid function, the hyperbolic tangent function, the rectified linear unit (ReLU) function, and its variants such as leaky ReLU and exponential linear unit (ELU). The activation functions can either be differentiable or non differentiable. The choice of activation function can have a significant impact on the performance of a neural network, and appropriate consideration is given.



(a) Sigmoid function

(b) Hyperbolic tangent function

Figure 2.2: (a) the sigmoid function and (b) the hyperbolic function tanh both examples of differentiable, monotonically increasing functions.



(a) ReLU function

(b) Leaky ReLU function

Figure 2.3: (a) the rectified linear unit function and (b) the leaky rectified linear unit function, both examples of non-differentiable, monotonically increasing functions. Non-differentiable functions can create problems with learning, as numerical gradients calculated near a non-differentiable point can be incorrect.

Supervised problems are categorised as classification or regression problems. Activation functions that are commonly used for classification problems belong to the sigmoid family. A sigmoid function is a bounded, differentiable, real function that is defined for all real input values and has a non-negative derivative at each point and exactly one inflection point.Common activation functions in this family are sigmoid, hyperbolic tangent and the softmax function.

For regression problems, the goal is to predict a continuous numerical output value. Typically, activation functions for regression problems have piecewise linearity, which further helps calculate a continuous numerical output. Common activation functions for regression problems are the linear activation function, ReLU, Leaky ReLU, and Softplus. Note that the Tanh function is nonlinear but is symmetric around 0 and can handle negative values, making it useful for regression problems.

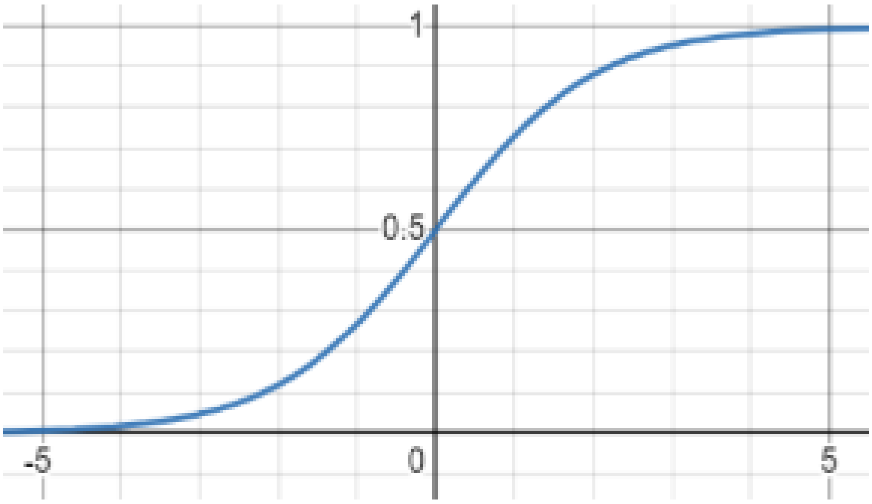


Figure 2.4: The sigmoid function compresses the output into the range [0, 1] most commonly used for binary classification, where the output can be interpreted as the probability of belonging to the positive class. This function has a bias towards 0 and 1.

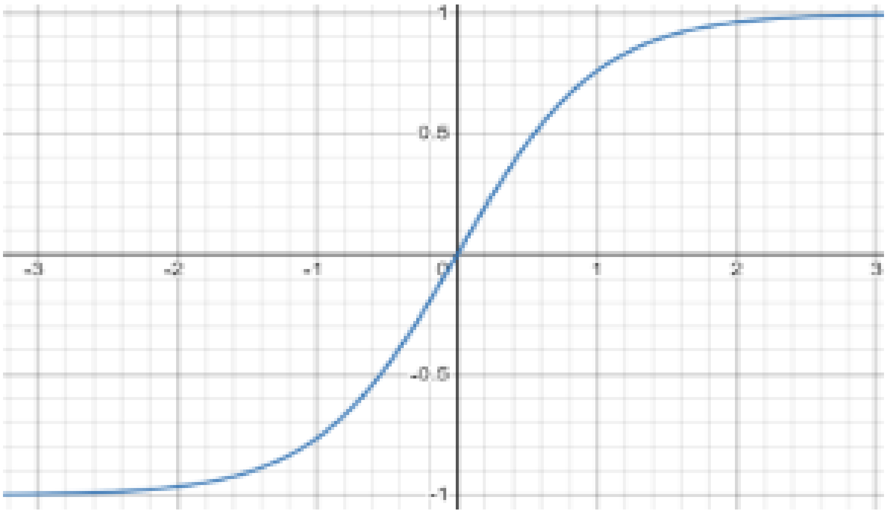


Figure 2.5: The hyperbolic tangent function tanh compresses the output into the range [-1,1] most commonly used for classification of zero-centred data or data that also contains negative values. This function has a bias towards -1 and 1.

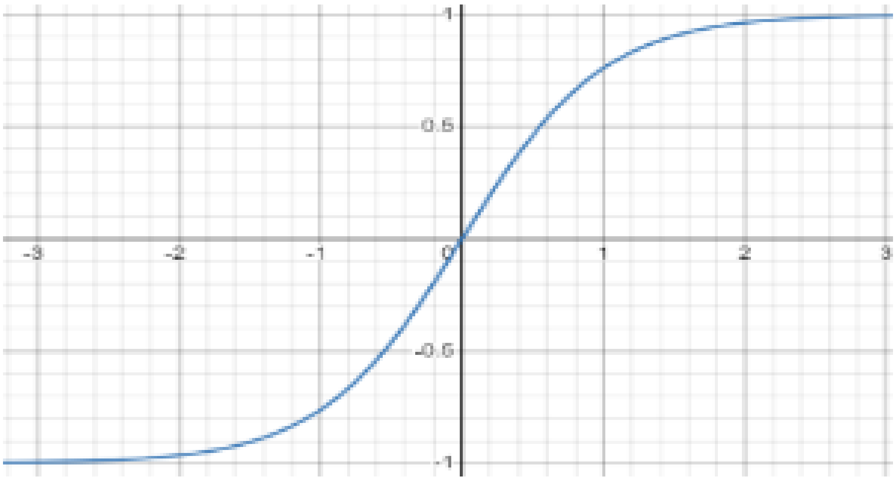


Figure 2.6: The softmax function takes as input a vector of real-valued scores, which can be seen as the evidence for each possible class, and outputs a probability distribution over the classes that sums to 1. The output of the softmax function can be interpreted as the model's confidence in the prediction for each class.

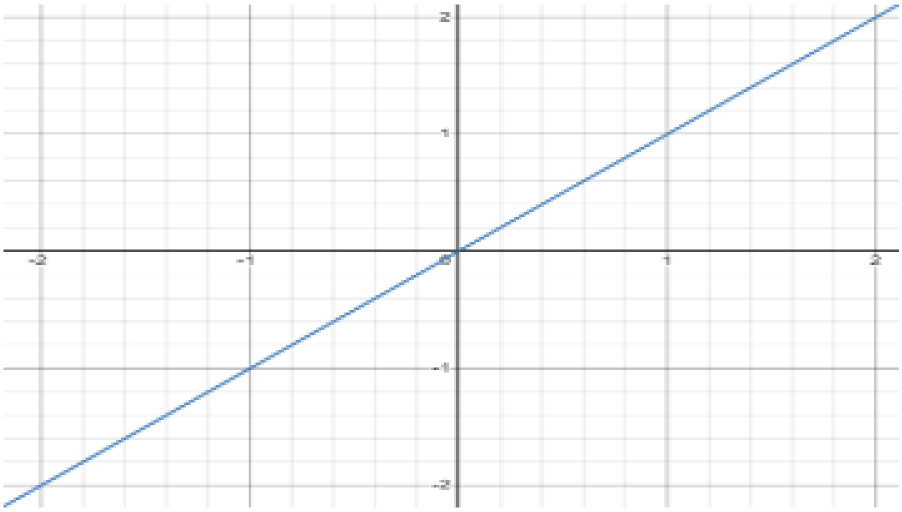


Figure 2.7: The output of the linear function is the input value multiplied by a constant factor and possibly shifted by another constant factor (bias), without any non-linear transformation. $output = (input \times weight) + bias$

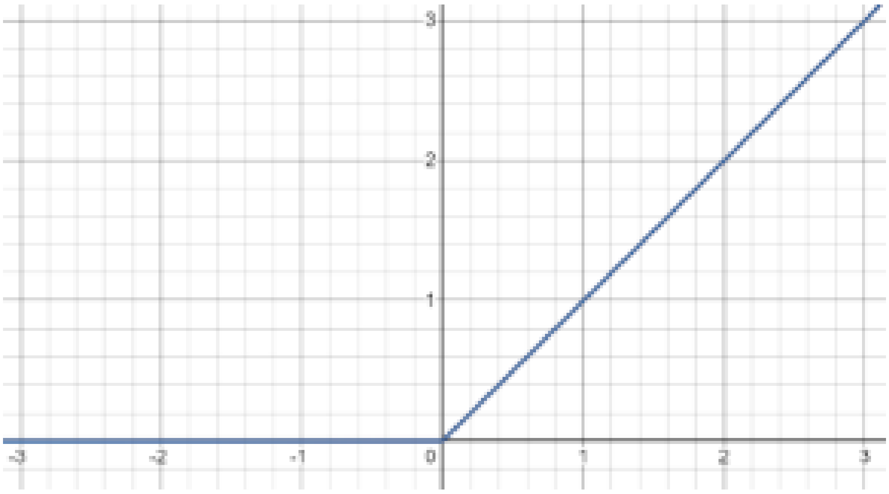


Figure 2.8: The ReLU function returns zero for all negative input values and the input value itself for all non-negative values. This can help introduce non-linearity in the network.

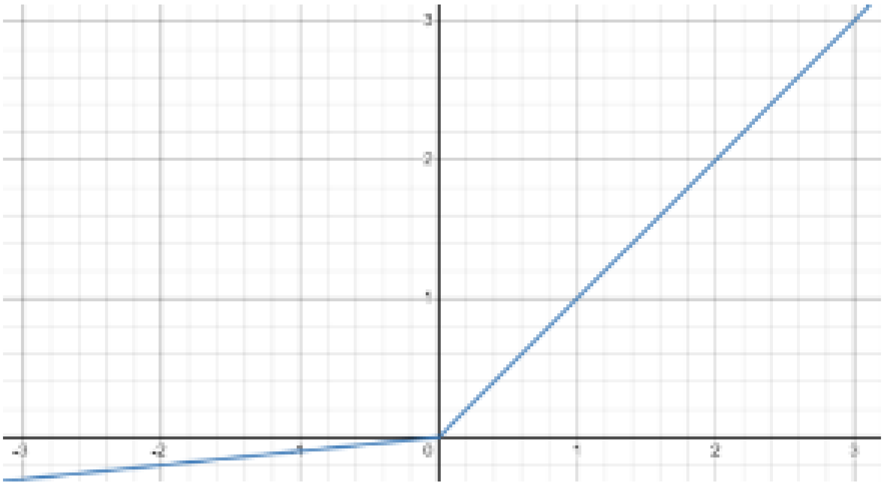


Figure 2.9: The Leaky ReLU is a variant of Relu, with the key difference being that for negative input, it doesn't output 0, but outputs a small constant multiplied by the input value. This prevents neurons from getting stuck with zero output and no gradient being propagated through them during backpropagation, leading to no updates of their weights. (The dying ReLU problem).

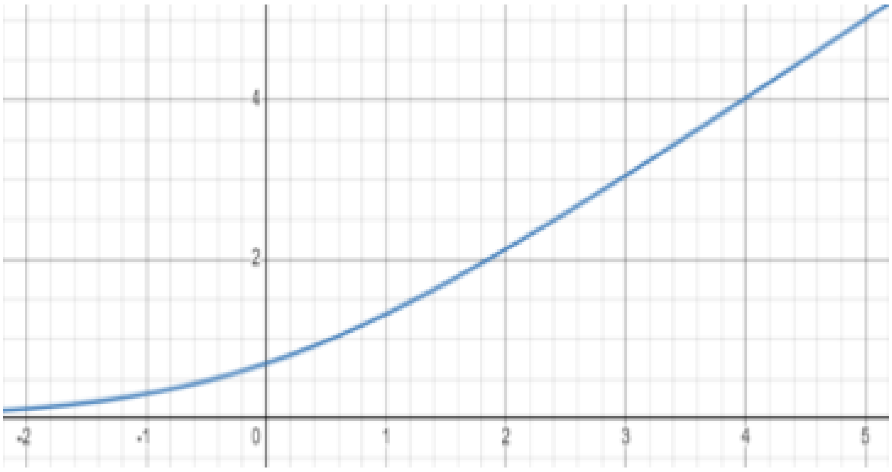
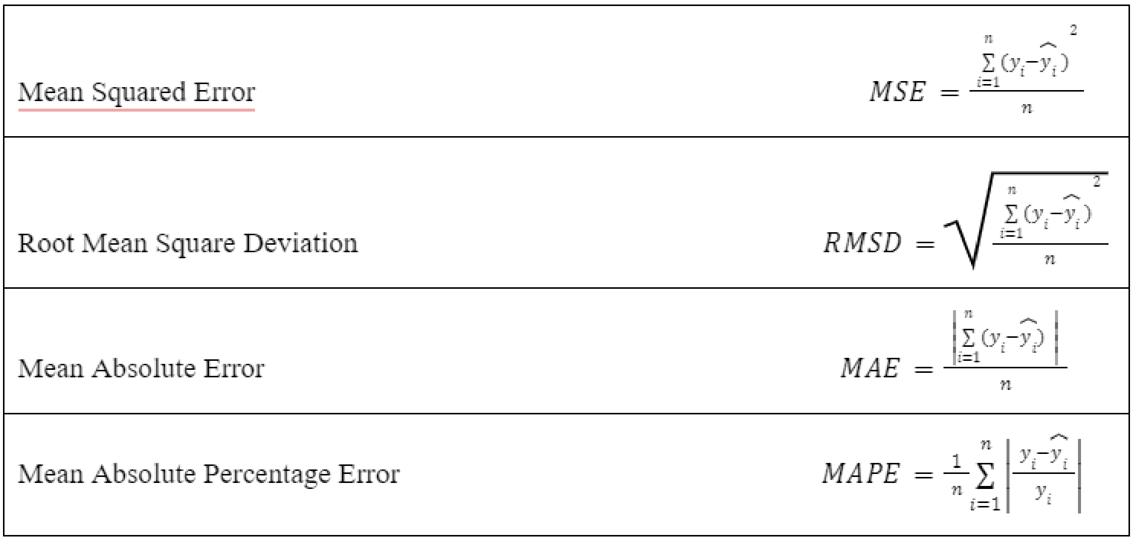


Figure 2.10: The Softplus activation function is similar to the ReLU activation function, but it is a smooth, differentiable function $f(x) = ln(1 + e^x)$

4. Cost functions

The cost functions are a crucial component in machine learning algorithms. They function as a metric of a model's accuracy and accommodate by guiding towards the best set of parameters that will minimise the error between the actual and predicted outputs during the model‘s training. Functioning as a representation of the error or cost associated with the prediction, the cost function uses as input the predicted output as well as the actual output and computes a value which the model attempts to minimise. The choice of cost function depends on the specific task and type of model being used. For example, in regression problems where the goal is to predict a continuous value, a common choice of cost function is the mean squared error (MSE). In classification problems where the goal is to predict a categorical label, a common choice of cost function is the cross-entropy loss, which measures the difference between the predicted probability distribution and the true distribution of labels. There exist numerous different cost functions, namely, the mean squared error function (MSE), mean absolute error (MAE), hinge loss, l1 loss, etc. The choice of cost function depends on the specific task and the model being used. For example, in regression problems where the objective is to predict a continuous value, a frequent choice of cost function is the MSE, which measures the average squared difference between the predicted and actual values. Similar to the activation function, the choice of cost function is subject to change, and commonly many different functions are applied, with the most efficient being applied to the final model.



Mean Squared Error	$MSE = \frac{\sum_{i=1}^{n}(y_i - \widehat{y_i})^2}{n}$
Root Mean Square Deviation	$RMSD = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \widehat{y_i})^2}{n}}$
Mean Absolute Error	$MAE = \frac{\left\lvert\sum_{i=1}^{n}(y_i - \widehat{y_i})\right\rvert}{n}$
Mean Absolute Percentage Error	$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left\lvert\frac{y_i - \widehat{y_i}}{y_i}\right\rvert$

Figure 2.11: Commonly used cost functions. Notations: (a) y_i = observed value, (b) $\hat{y_i}$ = estimated value, (c) n= number of data points, (d) i = iterative variable i.

5. Backpropagation

Backpropagation is the algorithm by which the scalar value of the cost function is minimised. It is a method for computing the gradient of the loss function with respect to the weights of the neural network. Commonly, by using gradient descent or other methods, the network reaches an optimised state. The backpropagation algorithm is based on the chain rule of calculus, which allows the computation of the derivative of a composite function. For example, if the inputs and outputs of the model are a multi-dimensional array of numerical data (tensors) Ti[m,n] and To[j,k] each layer of the model can be seen as a function that transforms Ti to To. This means that the derivative of the cost function with respect to the weights can be computed by successively applying the chain rule backwards through the layers of the network. Firstly, the input data is passed into the layers of the model, and the output for each layer is calculated with its current weights. Secondly, using the chosen cost function, the error is calculated. Then, the derivative of the loss with respect to the weights of each layer is calculated using the chain rule, starting from the output layer. Lastly, the weights are updated using the computed gradients. This process is repeated iteratively until the network converges on a set of weights that will minimise the loss function.

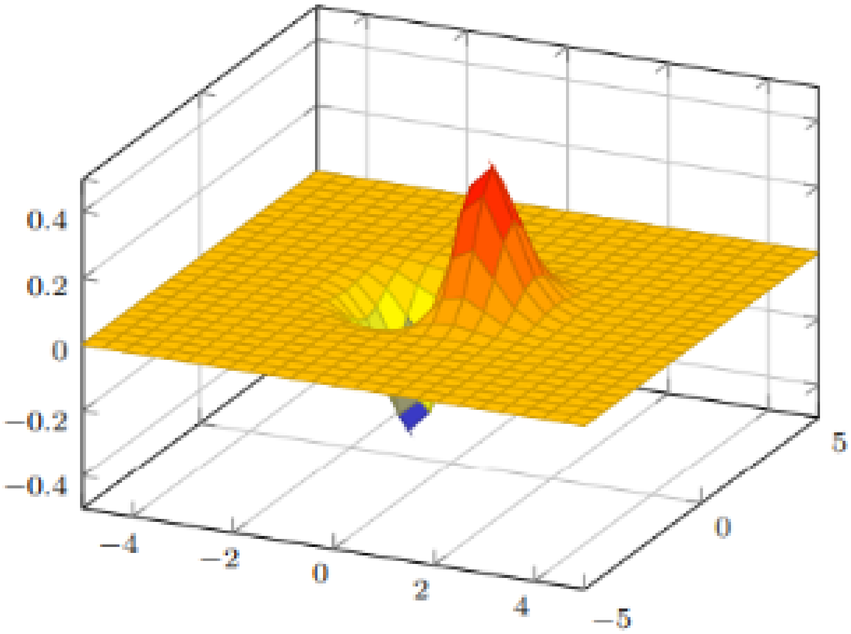


Figure 2.12: A 3d representation of the cost function, the goal of backpropagation is to guide the model to reach the lowest point with regards to cost.

6. Sequential Neural Networks

Sequential neural networks are a type of artificial neural network where the layers of the model are organised in a serial manner. In other words, the output from one layer is fed as input to the next layer, and so on, while the order of the layers stays fixed. As a neural network, it consists of an input layer, hidden layers, and an output layer. The information travels in one direction, from the input layer to the output layer. Each layer processes the information from the previous layer and passes it forward sequentially. As models, they are easy to design and interpret and can model complex nonlinear relationships between the input and output data, making them useful in applications such as speech and image recognition, natural language processing, predictive modeling, and time series analysis.

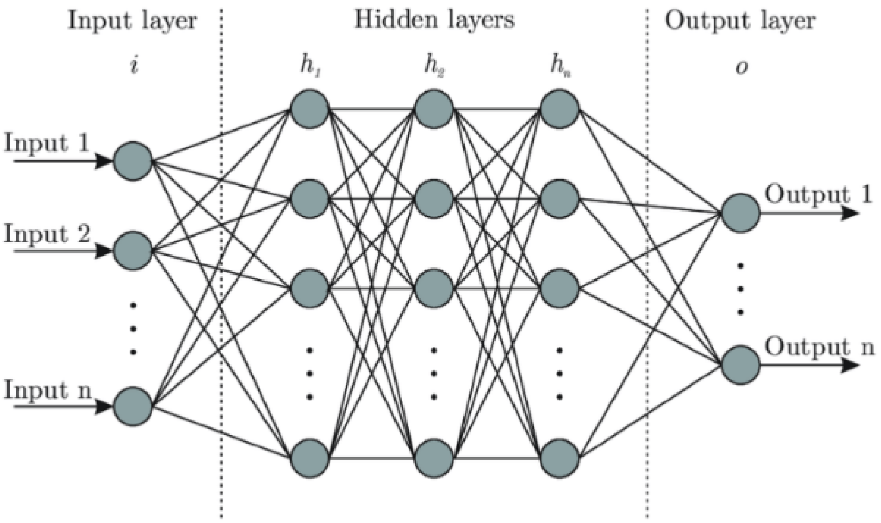


Figure 2.13: A Simple Sequential Neural Network.

Chapter 3

Data Analysis and Visualization of Trajectories

Trajectories are a key concept in machine learning. Understanding the trajectories of objects can help in predicting the behaviour of complex systems and optimising the performance of machine models. Geographical trajectories can inform policy and decision-making in areas such as urban planning, transportation, and emergency response. Mostly, they are influenced by a range of factors, including geography and climate. In order to study geographical trajectories, extremely large datasets need to be analysed (Big Data) and various tools and techniques are applied, namely visualisation tools such as Geographic information system (GIS) and Automatic Identification System (AIS).

1. Big Data

Big Data are extremely large datasets that are too complex or varied to be analysed using traditional data processing tools. They can be generated from various sources, such as, social media, online transactions, and sensor networks. The main features when mentioning big data are the 3 Vs (volume, velocity, and variety). Volume refers to the size of the data; velocity is the speed at which these data are generated and need to be processed; and variety refers to the diverse range of data types inside the dataset. Big data has become important in recent years due to the rise of new technologies that allow the creation, manipulation, analysis, and processing of large datasets. The challenge of processing and analysing such datasets is the epicentre of many studies in the field of computer science.

2. Geographic Information System (GIS)

GIS stands for Geographic Information System. It is a powerful computer system used for capturing, storing, analysing, and displaying geographically referenced information. GIS software allows users to create maps, and perform spatial analyses to gain insights into complex phenomena and relationships between different geographic features. GIS data can be acquired from a variety of sources, such as satellite imagery, GPS data, and ground surveys. GIS can help with combining and integrating the data sources in order to create a comprehensive and detailed visualisation of a particular phenomenon in a geographic area. Overall, it is a powerful, versatile tool with a wide range of applications.

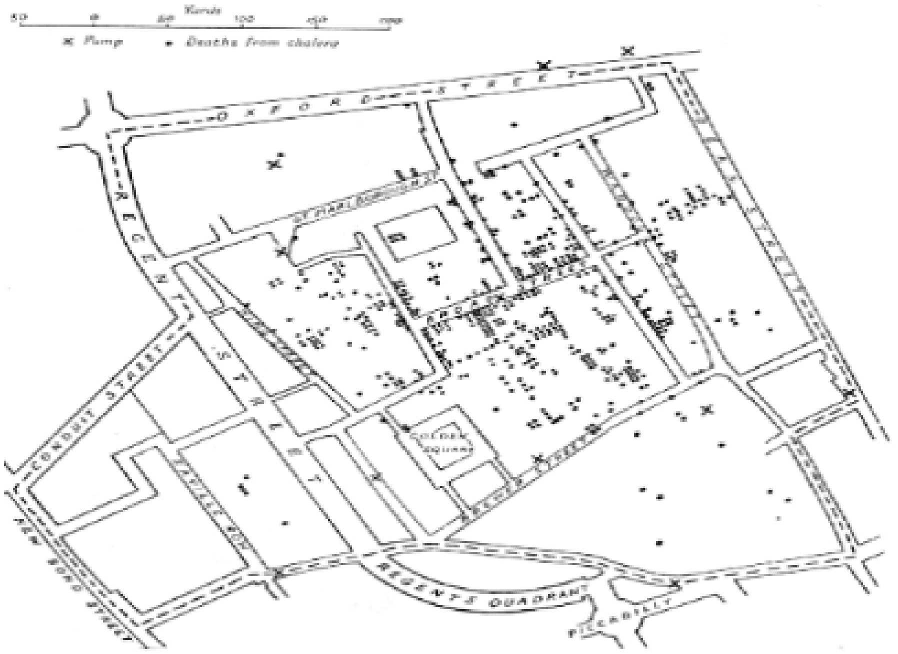


Figure 3.1: E. W. Gilbert's version (1958) of John Snow's 1855 map of the Soho cholera outbreak showing the clusters of cholera cases in the London epidemic of 1854 one of the earliest successful uses of a geographic information analysis.

3. Automatic Identification System (AIS)

The Automatic Identification System (AIS) is a technology used in the maritime industry that uses transceivers, radio signals, marine radars, and sometimes satellite signatures to track and identify vessels in the surrounding area and share this information with nearby vessels and authorities. The vessels using AIS broadcast real time information such as position, course, speed, and vessel identity, which is collected and sent to a central database to be accessed and used by authorised users. Its main uses are collision prevention, fishing fleet monitoring, accident investigation, and search and rescue missions. Furthermore, it can be used to monitor vessel traffic and improve shipping routes through the use of computer algorithms. Some potential limitations of AIS fall in the fields of data privacy, cyber attacks, signal jamming, and the possibility of data emission errors. However, AIS is an important technology that continues to develop in order to improve safety, efficiency, and sustainability. In conjunction with GIS, it is a powerful tool that can help create machine learning models for trajectory prediction, autonomous shipping and dynamic collision avoidance.

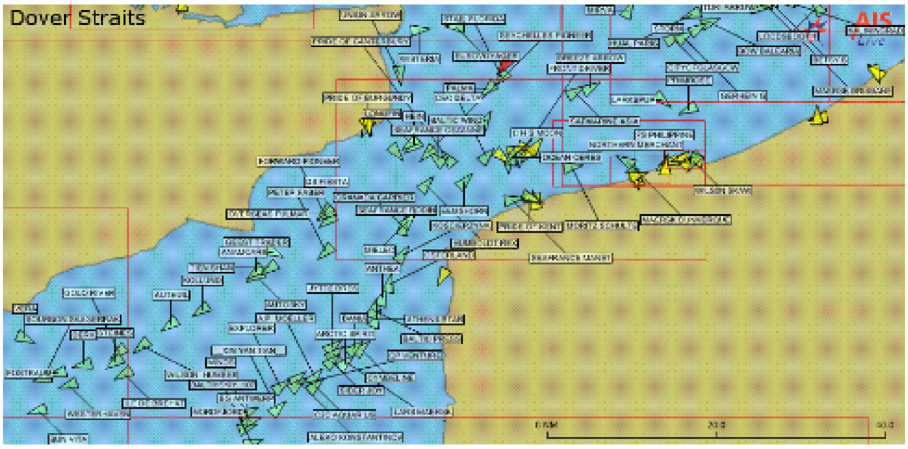


Figure 3.2: A graphical display of AIS data on board a ship.

Chapter 4

Literature Review

The Trajectory Prediction (TP) problem has received a lot of attention in recent years. This general problem can be further classified into two subfamilies: short-term prediction and long-term prediction [1, 2] [A Scalable Framework for Trajectory Prediction]. [Data-driven Driven Digital Twins for the Maritime Domain]. The focus of short term prediction models is on predicting the object's next location precisely, whereas long term prediction focuses on predicting possible next locations and routes. The proposed combating methods are mostly Machine Learning (ML) methods, hybrid in nature, and their components fall into three categories: rule-based prediction approaches, clustering-based prediction techniques, and predictive analytics methods. [3, 1][eva chon,An Efficient LSTM Neural Network-based Framework for Vessel Location Forecasting][A Scalable Framework for Trajectory Prediction].

1. Combating methods for the TP problem

There are several methods for combating the trajectory prediction problem in this section we will analyse each approach and provide their strengths and weaknesses.

1..1 Neural Networks (NNs)

A plethora of research has been done, expanding on the methods mentioned in Chapter 2. As proposed in [4, 5, 6] rigorous experiments have been made applying numerous different NN models. It is derived that using NN models provides us with high accuracy, especially when dealing with complex patterns and large datasets due to the non-linear relationship modelled between features in NN models. Additionally, NNs tend to be robust and able to handle noisy and incomplete data, making them ideal for handling data without the need for manual feature engineering after they learn the relevant features from raw sensor data. Neural networks can be categorised as static or dynamic. Static neural networks have a fixed architecture and the weights and biases of the network are pre-determined before the network is used for inference. In contrast, dynamic neural networks have an adaptive architecture where the network can change during inference based on the input data. In tasks with fixed input size, such as the one we are studying static neural networks are commonly preferred. Unless it is specified beforehand, from this point on, when a neural network is referred it is implied as a static NN. Below, we will expand on the tools used to create NNs for the Trajectory Prediction problem.

1..2 Rule-Based Methods

Using rule-based predictions is of great help in TP problems, especially when the nature of the problem requires limitations such as the road network and aircrafts moving in airway space. Sadly, aquatic vessels sail with a 2D flexibility that is derived from the non-fixed and sparser seaways. [3] [An Efficient LSTM Neural Network-Based Framework for Vessel Location Forecasting] Although rules are crucial for predictions, their application to aquatic vessels gets limited due to the nature of maritime travel.

1..3 Clustering-Based Methods

Further facilitation of a NN to increase accuracy usually comes in the form of clustering. Especially in aquatic vessels where rule based methods are limited; clustering based methods have a greater role in the improvement of the model. Clustering based methods sacrifice time at the initial stages of building the NN models in order to properly feed the model with well clustered data that will raise the model‘s accuracy. Notable methods used for clustering are Traj-clusiVAT [1][A Scalable Framework for Trajectory Prediction] where the technique was created and used to handle big data to create a scalable trajectory classification technique used for both short-term and long-term predictions. DBSCAN is also a commonly used technique, as seen in [2][Data Driven Digital Twins for the Maritime Domain] where after clustering and creating base trajectories, the DBSCAN algorithm was implemented on all the data points to create multiple path corridors derived from representative trajectories. As made clear in [2, 4][Data Driven Digital Twins for the Maritime Domain] and [i4sea] the knowledge of both trajectories and common sub-trajectories is used to extract crucial data that can further facilitate the NN models. Furthermore, a clustering method like TRACLUS [7][Trajectory Clustering: A Partition-and-Group Framework] can create simplified representative trajectories from all the data points and find common sub-trajectories based on trajectory line density. In this study, the methodology will delve deeper into the TRACLUS method, which will be the primary area of focus.

1..4 Predictive analytics methods

Predictive analytics can be used to analyse big data streams and predict, based on historical data, the future behaviors of objects. In [8][Employing traditional machine learning algorithms for big data streams analysis: The case of object trajectory prediction] the importance of feature selection and data sampling using decision Trees random forest and support vector machines is highlighted in order to improve the performance of models. In [3][An Efficient LSTM Neural Network-based Framework for Vessel Location Forecasting], the predictive analytics framework based on LSTM NNs can be used in a variety of applications, such as maritime traffic management, ship routing, and marine safety. The authors highlight its ability to predict future locations of vessels based on learned patterns and compare it with other machine learning models to demonstrate its superiority in terms of efficiency and accuracy. Predictive analytics are also used in [4][i4sea: a big data platform for sea area monitoring and analysis of fishing vessels activity] in the form of tools such as heat maps and trajectory analysis, to make predictions from data derived from big data technologies, such as Hadoop and Spark, using machine learning algorithms, like random forest and gradient boosting to predict the likelihood of illegal fishing. The i4sea platform is a practical application of predictive analytics that can be also applied in many marine conservation and management domains.

1..5 Markov Models

Markov Models(MMs) are a type of probabilistic model that can be used for sequence prediction tasks such as speech recognition and natural language processing[9][S. Haykin, Neural Networks: A Comprehensive Foundation, 2nd ed.USA: Prentice Hall PTR, 1998.]. Existing methods for the TP problem are based mostly on data-driven methodologies, such as Hidden Markov Chains and Neural Networks (NN) [3][An Efficient LSTM Neural Network-based Framework for Vessel Location Forecasting]. An example of Markov models can be seen in [1][A Scalable Framework for Trajectory Prediction] where a combination of MMs and NN models were used to predict trajectories of objects such as vehicles, pedestrians, and animals. To address the limitation of existing approaches they used a hierarchical approach that combines MMs and NNs. the MMs capture short-term dependencies in the trajectory data, while the NNs capture long-term dependencies thus providing more accurate predictions overall. A similar approach can be seen in [10][Trajectory Forecasting With Neural Networks: An Empirical Evaluation and A New Hybrid Model] where the study proposes a hybrid model that combines a feedforward neural network with a Markov model. The neural network is used to capture long-term dependencies in the trajectory data, while the Markov model is used to capture short-term dependencies.

2. Drawbacks

Although the methods referenced in chapter 4.1. are used due to their competence and efficiency in solving the TP problem, this section will focus on and analyse their general drawbacks.

2..1 Drawbacks of Machine Learning models

Under certain circumstances, ML models are prone to underperform when the data quality or quantity is subpar. This often happens when the data used is noisy or incomplete. Extensive care must be taken by the researcher to avoid using data that is not fit for a ML model. In some cases, the models created can be difficult to interpret, making it challenging for the researcher to understand why the model is making certain predictions, which is especially problematic in domains where the reasoning behind the prediction needs to be understood. In regards to the training process, special attention is given to avoid overfitting, which can lead to poor performance on test data and an inability to generalise on new data. During the model-designing process, the training data is extensively checked to avoid bias and address fairness concerns to ensure that the resulting model is developed in a way that is fair and unbiased. Lastly, the algorithm selection process can be challenging, and selecting the appropriate one can be difficult since different algorithms have different strengths and weaknesses, and a poor choice can lead to poor results.

2..2 Drawbacks of Clustering based techniques

Clustering-based techniques are often based on distance measures and are sensitive to noise and outliers, especially if those are not handled correctly. Additionally, their simplistic nature makes them unable to capture complex relationships between variables, which can cause underfitting. They usually require a posteriori knowledge of the optimal number of clusters [3][An Efficient LSTM Neural Network-based Framework for Vessel Location Forecasting], which may not be known a priori, and an incorrect choice of cluster number can lead to inaccurate predictions. In regards to clusters, points in the same cluster are treated as similar between them and dissimilar with points in other clusters, which, in practice, may not be the case since clusters can be overlapping or can have non-uniform densities. Lastly, they are not well suited for time-series problems due to their lack of understanding of temporal dependencies between data points, leading to inaccurate predictions when the nature of the data contains complex patterns or trends.

2..3 Drawbacks of Predictive analytics methods

Predictive analytics methods, such as Markov Models and hidden Markov Models, have proven to be very useful in making predictions, but their predictive ability is confined to the data that is used to train the model, giving them a limited scope [10][Trajectory Forecasting With Neural Networks: An Empirical Evaluation and A New Hybrid Model]. If the data is not representative of the general population, the predictions will not be accurate. Furthermore, it is important to note that regression models can only identify correlations between variables, not causation. Thus, predictions by the model should be interpreted with caution, as there may be underlying variables that are not captured in the model. Finally, variables with nonlinear relationships are not properly understood by models using predictive analytics in view of the fact that they assume linearity between input variables.

2..4 Drawbacks of Markov models

Specifically, Markov models assume that the probability of an outcome depends on the previous state, disregarding more complex relationships of higher order between variables. This in turn shows the model's dependency on the initial conditions; if the initial conditions are not accurately known, the predictions are unreliable. Even with those factors accounted for, Markov models do not have the ability to update the probability distributions of their states over time, and they assume that the future is independent of the past, being unable to take into account the history of the process beyond its current state [10][An Efficient LSTM Neural Network-based Framework for Vessel Location Forecasting]. Lastly, their dependability on large sample sizes to estimate the transition possibilities between states makes them unreliable for smaller sample sizes.

2..5 Drawbacks of rule-based algorithms

In general, applying pre-defined rules to the data can make the algorithm inflexible and unable to adapt to changes in the data or new scenarios. Also, the complexity of the rules can make it challenging to interpret the predictions or identify the rules that contribute the most to the predictions. Lastly, applying rule based algorithms implies that the rules applied capture all the relevant features of the data, which can be difficult to estimate in complex systems and can create an inability to handle noise and outliers, which sequentially impacts the performance of the model.

Chapter 5

Methodology

In this study we developed a method to predict future trajectories of maritime vessels by applying a clustering algorithm that clusters similar trajectories together and then training accordingly a feed forward neural network that predicts future positions of vessels based on the cluster they belong to. To evaluate the results we compare the methods accuracy with the results of a sequential neural network without clustered data. The following sections describe in detail the data, tools, and techniques used in this study to carry out trajectory prediction with the use of NNs and trajectory clustering with the TRACLUS algorithm[7].

1. The data

The data used are from passenger ships for the month of January 2018. The focus will be on the Aegean Sea. Identifying the longitude and latitude values that will be the boundaries of the region in question will help clean the data from outliers. The boundaries of the Aegean Sea are not fixed and can vary depending on the source of information. Arbitrarily the rectangle created from the coordinates 35.7897° N, 39.0713° N, 22.2140° E and 28.2792° E encompasses the Aegean Sea and parts of the surrounding landmass.

2. About Data preparation

Data preparation/Wrangling refers to the process of preparing data for training and testing machine learning models. This includes tasks such as data cleaning, data transformation, data encoding, and data splitting. The goal of data preparation for machine learning is to ensure that the data used to train the machine learning model is of high quality and can be effectively used to learn patterns and relationships in the data. Data preparation is considered one of the most time-consuming and crucial steps in the life cycle of a ML project. Given a raw dataset, data preparation involves the process of removing errors,checking for inconsistencies and in general ensuring reliability, accuracy and consistency across the dataset.

Table 5.1: Sample Preprocessed Data [10.283.171 rows x 8 columns].

mmsi	timestamp	lon	lat	shiptype	speed	course	heading
355931000	2018-10-02 20:11:12	23.870230	37.402481	60	174	341	341.0
237233600	2018-10-02 20:11:14	25.740101	35.007332	65	0	359	511.0
237294700	2018-10-02 20:11:16	23.542440	37.959049	60	71	86	511.0
271041325	2018-10-02 20:11:16	27.429489	37.034210	60	0	0	511.0
271002606	2018-10-02 20:11:16	27.954029	40.355400	60	0	0	511.0
...	...	...	...	...	...	...	...
237641000	2018-10-02 20:10:56	23.869181	37.349380	69	200	157	151.0
249727000	2018-10-02 20:11:01	24.457350	38.598270	60	141	316	316.0
237032000	2018-10-02 20:11:07	24.736830	35.937832	60	206	327	328.0
271044139	2018-10-02 20:11:07	28.730591	40.890369	60	108	323	511.0
239575000	2018-10-02 20:11:07	23.834829	37.381168	60	188	158	158.0

2..1 Data preparation

Since a regional window was already chosen, the first step will be to delete all rows from the dataset that are beyond the window that is allowed. regarding the homogeneity and standardization of the data, care is given to every column and duplicate rows are deleted. e.g. timestamps are converted to unix integers with a datetime pandas conversion, and longtitude and latitude degrees are turned to WGS84 decimals. Columns that are considered unnecessary, or do not inspire confidence will be deleted e.g. speed column. Lastly, columns that will be helpfull later on are added .e.g. the column dt which denotes how much time has passed from one signal to the next, and the new speed column.

2..2 Further Data preparation

After completing the basic Data wrangling mentioned, further preparation is applied to the data. The trajectories of the ships, that are denoted from their MMSI identification are splitted based on logical assumptions. A trajectory cannot be less than 10 timesteps or more than 1.000 timesteps and if there is a gap between signals more than 30 minutes the trajectory is split at that point. Furthermore a trajectory cannot last more that 24 hours. Regarding the vessels' speed, if the speed is less than 0.1 nautical knots its considered stopped, the trajectory is split and the row is deleted. Finally if the speed is greater than 25.7 nautical knots its consider unrealistic and the trajectory is split. This process is iterative and continues until no further changes are made by the program. For the final step, basic data analysis is applied and the dataset is shuffled to avoid overfitting when training and is split in 3 parts for training, validation and testing. Overall, data cleaning is a critical step in the data analysis process that ensures that the data is reliable, accurate, and consistent, and can be used for further analysis or modeling. Data cleaning requires careful attention to detail and a solid understanding of the data and the problem being solved.

mmsi	tr1_val2_test3	id	WGS84lon	WGS84lat	t	lon	lat	dist_m	dlon	dlat	dt	speed
237018400	1	1	23.530130	39.166431	1527491036	200205.617017	4.340983e+06	1086.690420	869.729229	651.511426	171.0	6.354915
237018400	1	1	23.538771	39.171829	1527491194	200975.271999	4.341554e+06	958.158259	769.654983	570.699970	158.0	6.064293
237018400	1	1	23.549900	39.178848	1527491404	201966.648647	4.342296e+06	1238.622488	991.376647	742.534856	210.0	5.898202
237018400	1	1	23.558029	39.183849	1527491555	202690.132658	4.342825e+06	895.938895	723.484012	528.467018	151.0	5.933370
237018400	1	1	23.569630	39.191471	1527491774	203724.494176	4.343633e+06	1312.623793	1034.361518	808.132089	219.0	5.993716
...	...	...	...	...	...	...	...	...	...	...	...	...
240080500	3	53715	23.543800	37.959721	1524390974	196343.777405	4.206983e+06	403.029610	-400.377580	-46.159074	123.0	3.276663
240080500	3	53715	23.538919	37.959450	1524391102	195913.693696	4.206969e+06	430.315890	-430.083709	-14.133955	128.0	3.361843
240080500	3	53715	23.534700	37.960499	1524391227	195547.244059	4.207099e+06	388.906798	-366.449637	130.242700	125.0	3.111254
240080500	3	53715	23.531450	37.964001	1524391362	195276.115323	4.207499e+06	482.708402	-271.128736	399.370267	135.0	3.575618
240080500	3	53715	23.530300	37.965599	1524391501	195181.669898	4.207680e+06	204.291116	-94.445425	181.148894	139.0	1.469720

Table 5.2: The data after processing

2..3 Converting timeseries to a supervised learning problem

Given a timeseries with 3 variables X,Y,Dt where X is the longtitude,Y is the latitude and Dt is the time difference from one timestep to the next, the goal is to create $n \times 3$ new columns where n is the number of lagged timesteps we want to create. Every new column will be the same with its predecessor but shifted by 1.

Table 5.3: Example mock Timeseries problem.

Longtitude X	Latitude Y	Time Dt
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	9	9

Table 5.4: Example mock Timeseries problem converted to supervised problem of lag timesteps 2.

Longtitude X_{-1}	Latitude Y_{-1}	Time Dt_{-1}	Longtitude X	Latitude Y	Time Dt
1	1	1	2	2	2
2	2	2	3	3	3
3	3	3	4	4	4
4	4	4	5	5	5
5	5	5	6	6	6
6	6	6	7	7	7
7	7	7	8	8	8
8	8	8	9	9	9
9	9	9	Nan	Nan	Nan

For the purpose of this research a step of $n = 10$ is used resulting in 156 columns. At this point, the dataset is checked to avoid rows that contain trajectory identities that do not match. After further cleaning and removal of redundant columns the dataset left contains ,in total, 66 columns. This technique is commonly used for machine learning, converting the dataset to a tabular form helps create a table where every row contains information of the object for the last n observations. The rows of the dataset that contain missing values are removed and the resulting table is split to two tables, one of observations $(t-10)...(t-9)$ and one of observations (t). The first will be uses as the input of our algorithm while the latter will be the goal output of the supervised problem. this tabular conversion is a powerfull tool that helps in making accurate predictions based on past observations.

Table 5.5: The input of the algorithm.

dlon(t-10)	dlat(t-10)	speed(t-10)	dt(t-9)	dlon(t-9)	...	dt(t)
678.048594	-265.691638	4.887556	141.0	616.915065	...	131.0
616.915065	-296.483109	4.854331	220.0	616.915065	...	131.0
955.288911	-440.407878	4.781455	131.0	577.215446	...	60.0
577.215446	-233.752623	4.753819	140.0	615.886767	...	220.0
615.886767	-211.968419	4.652447	149.0	606.261439	...	180.0
...	...	...	...	...	...	...

Table 5.6: The goal output of the algorithm.

dlon(t)	dlat(t)
676.550407	-198.792543
633.754767	-178.405028
592.817516	-164.923137
582.374405	-194.422268
616.910006	-206.865550
...	...
27.187918	-996.591755
-62.082231	-992.105461
-357.689199	-1269.006625
-473.783384	-622.380551
-573.308693	-307.355356

Lastly the two datasets created are split in training, validation and testing sets to further accomodate in solving the supervised problem. The training set will be the set on which the model will train its parameters while the validation dataset will be the control set that judges the accuracy and helps avoid overfitting, lastly the testing set will be the dataset that will judge the general accuracy of the model with data it has not seen yet. Of the original set, 55% will be used as the training set, 15% will be used as the validation set and 30% will be used as the testing dataset.

3. The Neural network model architecture

When attempting to create a neural network model various parameters have to be considered. A dense sequential neural network was chosen due to its simplicity, training speed and flexibility. The architecture of the model consists of an input layer with 42 variables, 4 dense hidden layers with population 32, 16, 8, and 4 neurons respectively, and an output of 2. Various layer compositions where tried. This decreasing layer approach was chosen to extract increasingly abstract features from the input data making it effective in capturing important feature points while minimizing the risk of overfitting. The data where scaled before passing through the model depending on the activation function used each time, best results were given when using a standard scaling of range [0, 1] and the ReLU activation function for every layer.Regarding the hyper parameters of the model, the chosen cost function was the mean squared error (MSE) while the optimising method used was an extended version of the stochastic gradient descent called Adam optimiser. Although Adam optimiser requires more computational power it has some benefits making it a better choice for the problem at hand. Some benefits of the Adam optimiser against the stochastic gradient descent are its ability to use adaptive learning rates for each weight and its ability to use both gradient and momentum to update the weight parameters. Finally, regarding the iterations of the training process, two methods were used. A check mechanism that compares the accuracy of the validation data of the current epoch with the validation accuracy of the previous accuracy, if the new weights are more accurate they are saved; And a patience parameter of size 10 that checks if any meaningfull uimprovement has been made, if for 10 consecutive epochs no meaningfull improvements are made the iteration stops. The model showed best results in the range of 18 to 25 iterations.

FTIAXE FWTO TO NEYRWNIKO SOU SE KANA PAINT. FIGURE THE NEURAL NETWORK MODEL.

4. Representative trajectories

olo to paper tou traclus oooooooofffffffffff oooooooofffffffffff oooooooofffffffffff oooooooofffffffffff

4..1 clusters of representative trajectories

4..2 Representative trajectories of the clusters

5. Training the Neural network model with clustered data

Chapter 6

Results and analysis

Chapter 7

Conclusion

Bibliography

[1] Punit Rathore, Dheeraj Kumar, Sutharshan Rajasegarar, Marimuthu Palaniswami, and James C. Bezdek. A scalable framework for trajectory prediction. *IEEE Transactions on Intelligent Transportation Systems*, 20(10):3860–3874, 2019.

[2] Alexandros Troupiotis-Kapeliaris, Nicolas Zygouras, Manolis Kaliorakis, Spiros Mouzakitis, Giannis Tsapelas, Alexander Artikis, Eva Chondrodima, Yannis Theodoridis, and Dimitris Zissis. *Data Driven Digital Twins for the Maritime Domain*. 08 2022.

[3] Eva Chondrodima, Nikos Pelekis, Aggelos Pikrakis, and Yannis Theodoridis. An efficient lstm neural network-based framework for vessel location forecasting. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–17, 2023.

[4] Panagiotis Tampakis, Eva Chondrodima, Aggelos Pikrakis, Yannis Theodoridis, Kostis Pristouris, Harry Nakos, Eleni Petra, Theodore Dalamagas, Andreas Kandiros, Georgios Markakis, Irida Maina, and Stefanos Kavadas. Sea area monitoring and analysis of fishing vessels activity: The i4sea big data platform. In *2020 21st IEEE International Conference on Mobile Data Management (MDM)*, pages 275–280, 2020.

[5] Eva Chondrodima, Petros Mandalis, Nikos Pelekis, and Yannis Theodoridis. Machine learning models for vessel route forecasting: An experimental comparison. In *2022 23rd IEEE International Conference on Mobile Data Management (MDM)*, pages 262–269, 2022.

[6] Petros Mandalis, Eva Chondrodima, Yannis Kontoulis, Nikos Pelekis, and Yannis Theodoridis. Machine learning models for vessel traffic flow forecasting: An experimental comparison. In *2022 23rd IEEE International Conference on Mobile Data Management (MDM)*, pages 431–436, 2022.

[7] Jae-Gil Lee, Jiawei Han, and Kyu-Young Whang. Trajectory clustering: A partition-and-group framework. SIGMOD '07, page 593–604, New York, NY, USA, 2007. Association for Computing Machinery.

[8] Angelos Valsamis, Konstantinos Tserpes, Dimitrios Zissis, Dimosthenis Anagnostopoulos, and Theodora Varvarigou. Employing traditional machine learning algorithms for big data streams analysis: The case of object trajectory prediction. *Journal of Systems and Software*, 127:249–257, 2017.

[9] Simon Haykin. *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1998.

[10] Yuan Wang, Dongxiang Zhang, Ying Liu, and Kian-Lee Tan. Trajectory forecasting with neural networks: An empirical evaluation and a new hybrid model. *IEEE Transactions on Intelligent Transportation Systems*, 21(10):4400–4409, 2020.

Appendix A

Το λογισμικό