Cairo University

The 56th Annual International

Conference of Data science

Computer Science

4 – 6, Dec, 2023

Index

Computer Science

Predicting pIC50 against SARS Coronavirus Using Machine Learning

Ghadeer Adel Ali Sabek, Zeinab E. Attia
Computer Science Department, Faculty of Graduate Studies for Statistical Research

Abstract:

The process of drug discovery, which involves identifying potential therapeutic agents, is a time-consuming and expensive undertaking. Recent advancements in Information and Communication Technologies and Machine Learning (ML) techniques have emerged as valuable tools in the pharmaceutical industry, facilitating accelerated and automated analysis of available data. In particular, ML algorithms play a crucial role in the development of Quantitative Structure Activity Relationships (QSAR) models. This study compares the performance of regression models based on molecular fingerprints and Lipinski descriptors in predicting the pIC50 values of molecules. The models were evaluated using evaluation metrics such as R^2 and PCC, noticing that R^2 gives more realistic overview of how the model really works, and hyperparameter tuning was performed for the best-performing model. The optimized model was then used to predict the pIC50 values of input molecules, and a threshold of PIC50 $\geq$ 4.5 was employed to classify molecules as active or inactive. The results indicate that the fingerprint-based regression model outperformed the descriptor-based model, suggesting that fingerprints provide a more effective and reliable approach for predicting the pIC50 values of molecules. These findings contribute for decision-making in molecule selection for experimentation in drug discovery process.

Key Words: Drug discovery, Machine Learning (ML), Quantitative Structure Activity Relationships (QSAR), pIC50 prediction.

1.Introduction

Drug discovery is the process of identifying chemical entities that have the potential to become therapeutic agents. It is a time-consuming and costly endeavor [32]. Recent advancements in Information and Communication Technologies (ICT) and computational power have revolutionized drug discovery by enabling more efficient and cost-effective screening of vast libraries of drugs. Machine Learning (ML) techniques have emerged as valuable tools in the pharmaceutical industry, allowing for automated analysis and prediction tasks [6,7].In particular, Quantitative Structure Activity Relationships (QSAR) models have gained prominence as mathematical models that can predict the properties of chemicals based on their structural features [15].

The purpose of this study is to develop a QSAR model that assists medical researchers in the initial stages of drug discovery, via helping researchers in the selection of potential drug candidates that have the ability to inhibit the SARS Coronavirus, thereby providing valuable guidance for subsequent experimental investigations. The proposed model focuses on predicting pIC50 values, which measure the inhibitory potency against the SARS Coronavirus, for potential drug candidates. The study compares the performance of regression models based on molecular fingerprints and Lipinski descriptors, using evaluation metrics such as R^2 and PCC.

The contribution of this paper lies in demonstrating the superiority of fingerprint-based regression models over Lipinski descriptor-based models in predicting pIC50 values. Additionally, the study highlights the exceptional performance of the Gradient Boosting Regressor algorithm compared to other algorithms like Random Forest, Linear Regression, and Support Vector Regressor. The use of the evaluation metric R^2 provides a more accurate assessment of the regression model's performance.

The paper is organized into six main sections. The first section is the introduction as above. The second section provides an overview of drug discovery, QSAR, fingerprints, descriptors, and the concept of IC50. The third section explores machine learning concepts, algorithms, and popular libraries and toolboxes. The fourth section presents a review of related work from seven papers, providing a comprehensive background. The fifth section covers the data collection process, methodology and results. Finally, the sixth section which include conclusion and future work.

2.Drug Discovery

This section provides an overview of drug discovery and its phases, highlights the importance of QSAR in predicting the activity of novel molecules, discusses the use of descriptors and fingerprints in characterizing molecular properties, examines Lipinski's Rule of 5 as a guideline for drug design, and explains the use of IC50 and pIC50 values in evaluating drug efficacy. Understanding these concepts enhances our knowledge of drug discovery processes and the tools used in the field.

The process of drug discovery involves identifying new pharmaceutical drugs. It begins with basic scientists identifying a target structure associated with a disease and screening for lead compounds that have an affinity for the target [30]. The most promising leads undergo further development to assess their safety and effectiveness in model organisms and eventually in humans. Drug discovery is a complex and costly process, with an estimated cost of 2.6 billion USD to bring a new drug to the market [25].

QSAR is a mathematical modeling approach used to predict the physicochemical and biological properties of chemicals based on their experimental or calculated properties. QSAR enables researchers to establish a reliable quantitative relationship between the structure and activity of chemicals. Physicochemical properties refer to all the physical and chemical properties of a drug that invoke pharmacological response on the receptor, which can be a biological molecule or system with which it interacts. Biological properties correspond to investigating enzyme inhibitory activities, while structural properties pertain to atomic and functional constituents of molecules (i.e., molecular descriptors that describe molecular composition and chemical properties of compounds). It establishes a quantitative relationship between the structure and activity of molecules, allowing for the prediction of the activity of novel molecules before their synthesis. QSAR has evolved from basic regression and classification analyses to sophisticated machine learning-based techniques that can extract valuable information from complex molecular datasets. It has become an essential tool in drug discovery, enabling efficient and cost-effective prediction of activities and properties [20].

Descriptors and fingerprints are abstract representations of structural features of molecules used in QSAR analysis. Descriptors capture variations in the structural properties of molecules, while fingerprints provide more general representations that do not rely on pre-defined patterns. These representations help in characterizing molecular properties and are used as inputs in QSAR models. Lipinski's Rule of 5 (Ro5) is a set of guidelines developed in 1997 to assess the likelihood of a molecule's success as a drug. According to Ro5, molecules with more than 5 hydrogen bond donors, more than 10 hydrogen bond acceptors, a molecular weight greater than 500 Da, and a calculated Log P greater than 5 are less likely to be absorbed or permeate effectively. While Ro5 has

been useful in guiding drug design efforts, there is a need for innovation to engage newer targets for transformative medicines [16, 26].

IC50 is a widely used parameter for evaluating the effectiveness of a drug against specific biological targets [2]. It represents the minimum concentration of a drug required to inhibit the target's activity by 50%. Presenting IC50 values can be challenging due to variations in formats used in different studies. To address this issue, pIC50 values, which represent the negative logarithm of IC50, are used. pIC50 values facilitate the comparison of drug potency at the same molar levels and are widely employed in computer-aided drug design approaches [9, 1].

3.Machine Learning in Drug Discovery

The growth of Artificial Intelligence (AI) and Machine Learning (ML) has had a significant impact on various industries, including pharmaceuticals. ML is a process where computing systems learn from data and use algorithms to perform tasks without explicit programming. In the field of drug discovery, ML approaches have emerged as valuable tools for improving discovery and decision-making processes by leveraging high-quality data [21].The integration of AI and ML in drug discovery has the potential to enhance the effectiveness and precision of the drug development process. These technologies not only improve efficiency but also have the capability to replace or reduce the need for actual clinical trials through simulations [8,29]. By using AI and ML, researchers can conduct more extensive studies on molecules, leading to cost savings and addressing ethical concerns [21].

Several ML algorithms have gained prominence in drug discovery. Support Vector Machines (SVMs) are widely used for compound classification, searching for new active compounds, and predicting their properties. SVMs are supervised learning algorithms that aim to minimize generalization errors by computing a linear regression function within a high-dimensional feature space [12, 3,24]. Random Forests (RF) are another popular ML algorithm in drug discovery. RF combines multiple tree predictors, with each tree depending on values sampled from a random vector. RF is advantageous for both regression and classification tasks, as it overcomes overfitting issues by averaging predictions from multiple trees [5].Linear Regression is a statistical analysis technique that models the relationship between dependent and independent variables. It aims to identify a straight line that best approximates the relationship between the variables. Linear Regression is widely used in drug discovery to predict outcomes based on independent variables [13]. Gradient Boosting Regressor is an adaptive boosting method that combines several weak learners to boost algorithm performance. It builds an additive model using decision trees as weak learners and updates gradients iteratively. The learning rate and maximum depth of the trees are important parameters in Gradient Boosting Regressor [31].

In the context of drug discovery, various libraries and toolboxes support ML applications such as RDKit, Pandas and NumPy. RDKit is an open kit toolbox for cheminformatics that facilitates descriptor calculations and includes functionalities for molecular operations [23]. Pandas is a Python library providing data structures and analysis tools, particularly suited for working with tabular data[18]. NumPy is a fundamental package for scientific computing in Python, offering multidimensional array objects and various mathematical operations[11].

Thus the integration of AI and ML in drug discovery has revolutionized the industry by improving efficiency, reducing costs, and enabling more extensive studies on molecules. Support Vector Machines, Random Forests, Linear Regression, and Gradient Boosting Regressor are among the famous ML algorithms used in drug discovery. Libraries and toolboxes such as RDKit, Pandas, and NumPy provide essential functionalities for ML applications. By utilizing these ML techniques and tools, researchers can accelerate the drug discovery process, enhance decision-making, and minimize risks associated with clinical trials.

4. Previous Work

We conducted a literature search for this research using specific keywords, such as "Machine Learning" or "ML," and "QSAR", and "Drug Discovery" or "Drug Development." We searched in publicly available databases, including Google Scholar, ResearchGate, ScienceDirect, BMC Bioinformatics, and PubMed. After applying filters based on the relevance of the content, abstract, methodology, required prior knowledge, and publication date between 2019 and 2023, we identified approximately 7 studies that were suitable for inclusion in our literary review. The studies are categorized based on the type of model employed, namely classification or regression. It is noteworthy that regression models hold greater significance and provide more informative insights compared to classification models. This is due to the fact that regression models not only predict whether a molecule is active or inactive, but also provide the value of IC50. By predicting the IC50 value, decision-makers can effectively prioritize and identify molecules that are more worthy of experimental study. Therefore, regression models play a crucial role in narrowing down the selection of molecules for further investigation, offering valuable guidance for decision-making processes.

Malik et al. [17] conducted an extensive study centered on Alzheimer's disease research, with a specific focus on discovering inhibitors for acetylcholinesterase (AChE) and butyrylcholinesterase (BChE) as potential therapeutic interventions. The research aimed to differentiate active compounds from inactive ones using interpretable molecular descriptors and a machine learning algorithm. To achieve this, a non-redundant dataset comprising 985 compounds for AChE and 1056 compounds for BChE, sourced from the ChEMBL database, was utilized. The researchers employed a random forest algorithm to construct predictive models, and after evaluation, the Substructure Count fingerprint emerged as the most reliable descriptor. The models demonstrated a five-fold cross-validated Matthews correlation coefficient (MCC) of [0.76, 0.82] for AChE and BChE, respectively. In light of their findings, the researchers developed a publicly accessible web server named ABCpred, which can be accessed at http://codes.bio/abcpred/. It is important to note the evaluation of the classification model's performance was conducted using appropriate measures such as the confusion matrix, accuracy, and MCC. However, the study did not propose a method for predicting the IC50 value for each input SMILES representation, as its focus was primarily on classifying molecules as active or inactive.

Kwon et al. [15] presented a comprehensive ensemble method that utilized multi-subject diversified models and second-level meta-learning. The authors introduced a novel type of individual classifier, based on one-dimensional convolutional neural networks (1D-CNNs) and recurrent neural networks (RNNs), within a multi-subject comprehensive ensemble framework. They claimed that their approach outperformed both individual models and other ensemble methods. To validate their proposed method, the authors conducted experiments on 19 bioassays from the PubChem open chemistry database. Bioassays are biochemical tests used to assess the potency of chemical compounds on specific targets, and they serve various purposes, such as drug development and environmental impact analysis. Subsequently, the authors applied their proposed model to a dataset related to HIV classification, aiming to distinguish between active and inactive compounds. They evaluated the model's performance using metrics such as F1 score, Matthews correlation coefficient (MCC), and confusion matrix, which are commonly used for assessing classification models. However, it should be noted that in three out of the 19 bioassays, the proposed model was outperformed by the random forest (RF) model. Additionally, RF consistently ranked among the top three models for most of the remaining bioassays. Moreover, to gain a better understanding of the proposed model's performance, it would have been beneficial for the authors to compare the model's performance with the RF and neural network (NN) models using the same dataset, as the RF and NN models were among the top three best-performing models. This would have allowed for a direct comparison of the MCC values for all three models.

Srivastava et al. [28] presented a research paper introducing the Molib tool, which aims to predict the biofilm inhibitory activity of small molecules. The authors emphasize the importance of identifying compounds that can inhibit biofilm formation in bacteria for potential therapeutic interventions. They note that experimental identification of such molecules can be time-consuming, and computational approaches like Molib offer promising alternatives. The researchers collected data primarily from the PubMed and KEGG databases.The Molib tool utilizes a carefully curated training dataset of biofilm inhibitory molecules and employs machine learning-based classification models. Various machine learning algorithms, including random forest (RF), support vector machine (SVM), CART, and kth nearest neighbor (kNN), were applied in the study. Among these algorithms, the random forest model demonstrated the best performance, achieving an ROC value of 0.96 for both descriptor-based and fingerprint-based predictions. Furthermore, the paper describes the construction of a hybrid dataset by combining the top 40 descriptors and 102 fingerprints. However, when evaluating the performance of the three models (descriptor RF, fingerprints RF, and hybrid RF) on a blind dataset, it was observed that the model based solely on descriptors performed the best, followed by the hybrid model, and finally the fingerprint model. Although Molib generally outperformed aBiofilm, another existing tool, the degree of superiority varied depending on the model employed (descriptor, fingerprint, or hybrid). The choice of using the ROC value as a means of comparing classifier performance is very convenient, as well as the use of the confusion matrix and Matthews correlation coefficient (MCC) to evaluate each model's performance. These metrics are suitable for assessing the classification capabilities of the models, as the model classify molecules as either biofilm inhibitory or non-inhibitory. However, the data used in the paper was not publicly available. Also, the tool could have been more useful if it can give the value of inhibition for each molecule instead of just classifying it as inhibitory or non-inhibitory as this helps in choosing which molecule, the one with lowest IC50, to start with as a potential drug target.

Soares et al. [27] made a valuable contribution to the field of anti-biofilm research, which focuses on combating the problem of antibiotic resistance caused by biofilms. Biofilms create a physical barrier that protects bacteria from the immune system and drugs, making them resistant to treatment. To address this issue, the researchers developed a machine learning technique called 'anti-Biofilm.' This technique uses a predictive algorithm to identify and analyze the effectiveness of small molecules in inhibiting biofilm formation. The algorithm was created using experimentally validated anti-biofilm compounds, and their inhibitory concentration values (IC50) were obtained from aBiofilm resource. The researchers employed five different machine learning models, namely Support Vector Machine (SVM), Random Forest (RF), Multilayer Perceptron (MLP), KStar, and M5Rules, to develop QSAR-based models. Among these models, SVM performed the best, achieving a Pearson's correlation coefficient of 0.75 on the training/testing dataset. To make their findings accessible, the researchers implemented the three most successful machine learning models (SVM, RF, and MLP) as a user-friendly web server called 'antiBiofilm'. This web server can be found at https://bioinfo.imtech.res.in/manojk/antibiofilm/.Generally the webserver is user friendly and allows users to input a molecule's SMILES representation and obtain the predicted PIC50 (-log (IC50)) value. However, it is worth noting that the paper used Pearson's correlation coefficient (PCC) for model evaluation, which may not be the most recommended approach. PCC is useful for identifying patterns but does not provide a comprehensive measure of the model's strength. In regression model evaluation, the coefficient of determination (R^2) is typically preferred. Additionally, since the dataset used in the study was not available to the public, it limits the reproducibility and further exploration of the findings.

Hammoudi et al. [10] presented a study focused on developing a QSAR model for the inhibition of the Acetylcholinesterase enzyme, which has potential therapeutic applications in treating Alzheimer's disease. The study specifically examined DL0410 and its 29 derivatives.The authors employed Multiple Linear Regression (MLR) analysis to construct the QSAR model. The model was found to be robust and demonstrated a high predictive capacity, as indicated by a coefficient of determination (R^2) value of 0.94 and a root-mean-square deviation (RMSE) value of 0.260. The use of R^2 and RMSE as evaluation metrics

for the regression model is very appropriate and suggests the model's robustness. However, it should be noted that the R^2 value decreased from 0.94 to 0.6 after applying cross-validation. Also, the paper did not specify the number of folds used in the cross-validation process. Additionally, the dataset used in the study was not available in the public domain, limiting the reproducibility and further exploration of the findings.

Rajput et al. [22] presented a paper highlighting the critical need for novel and highly effective biofilm inhibitors to combat antibiotic resistance. The authors developed a platform called "Biofilm-i" that utilizes a quantitative structure-activity relationship (QSAR) approach to predict the efficiency of chemicals in inhibiting biofilm formation. The dataset used for model development consisted of experimentally validated biofilm inhibitors obtained from the "aBiofilm" resource. The authors employed a 10-fold cross-validation approach and utilized machine learning techniques such as support vector machine and random forest to process the data. The Biofilm-i platform demonstrated strong predictive performance for various categories, including overall chemicals, Gram-positive bacteria, Gram-negative bacteria, fungus, Pseudomonas aeruginosa, Staphylococcus aureus, Candida albicans, and Escherichia coli. The integrated analysis tools of the platform allow for chemical structure conversion, searching for similar chemicals in the aBiofilm database, and analog design.The Biofilm-i platform is a valuable resource for researchers involved in designing effective biofilm inhibitors to combat antibiotic resistance. The platform can be accessed at https://bioinfo.imtech.res.in/manojk/biofilmi/ .The platform provides a user-friendly interface for inputting molecule SMILES and obtaining IC50 predictions. However, it is worth noting that within the paper, the authors mentioned R^2 (coefficient of determination), MAE (mean absolute error), or RMSE (root-mean-square error) and Pearson's correlation coefficient (PCC) as metrics to evaluate the regression model. Then, they only mentioned the result of Pearson's correlation coefficient (PCC) of the models. Additionally, the paper does not explicitly mention the availability of the data used in the study, which may limit the reproducibility and further exploration of the findings.

Khedekar et al. [14] presented a research study focused on utilizing a random forest regression model to predict the pIC50 values of novel drugs targeting acetylcholinesterase (AChE), a protein crucial for cognition and memory in Alzheimer's disease. The authors sourced training and test data from the ChEMBL dataset and employed RDKit and PaDEL Descriptor software to calculate the necessary descriptors and fingerprints. The evaluation of their model was conducted using Mean Square Error (MSE), R-squared (R^2), and Pearson Correlation Coefficient (PCC), providing a comprehensive assessment of the regression model's performance. Notably, the PCC was significantly higher than the R^2 score, indicating that the latter may be more suitable for providing a concise understanding of the model performance.

In conclusion, these studies demonstrate the application of machine learning and QSAR techniques in drug discovery. The development of web servers such as ABCpred, anti-Biofilm, Biofilm-I and Molib provides accessible platforms for researchers to utilize these predictive models. While the studies showcase promising results, it is important to consider the limitations of each approach, such as the availability of datasets and the choice of evaluation metrics. Also, regression models possess considerable importance and yield more substantial and informative insights in contrast to classification models. This distinction arises from the capability of regression models to not only determine the binary classification of molecules as either active or inactive but also provide a quantitative estimation of the IC50 value which can help to prioritize molecules that exhibit greater potential for experimental examination.

5. Methodology

5.1 Data Collection

Data collection is not an easy task when it comes to molecule bioactivity data especially for a new virus or disease, as the experimentally validated data are not very big. Data sets of inhibitors against SARS coronavirus (Target ID: CHEMBL3927) were compiled from the ChEMBL database [19]. Initially, there were a total number of 333 bioactivity data points (i.e. a heterogeneous mixture of bioactivity data reported in various bioactivity units including IC50, Ki, INH, KD, etc.) .

	cross_references	organism	pref_name	score	species_group_flag	target_chembl_id	target_components	target_type	tax_id
0	[]	Coronavirus	Coronavirus	17.0	False	CHEMBL613732	[]	ORGANISM	11119
1	[]	SARS coronavirus	SARS coronavirus	14.0	False	CHEMBL612575	[]	ORGANISM	227859
2	[]	Feline coronavirus	Feline coronavirus	14.0	False	CHEMBL612744	[]	ORGANISM	12663
3	[]	Murine coronavirus	Murine coronavirus	14.0	False	CHEMBL5209664	[]	ORGANISM	694005
4	[]	Human coronavirus 229E	Human coronavirus 229E	12.0	False	CHEMBL613837	[]	ORGANISM	11137
5	[]	Human coronavirus OC43	Human coronavirus OC43	12.0	False	CHEMBL5209665	[]	ORGANISM	31631
6	[{'xref_id': 'P0C6U8', 'xref_name': None, 'xre...	SARS coronavirus	SARS coronavirus 3C-like proteinase	10.0	False	CHEMBL3927	[{'accession': 'P0C6U8', 'component_descriptio...	SINGLE PROTEIN	227859
7	[]	Middle East respiratory syndrome-related coron...	Middle East respiratory syndrome-related coron...	9.0	False	CHEMBL4296578	[]	ORGANISM	1335626
8	[{'xref_id': 'P0C6X7', 'xref_name': None, 'xre...	SARS coronavirus	Replicase polyprotein 1ab	4.0	False	CHEMBL5118	[{'accession': 'P0C6X7', 'component_descriptio...	SINGLE PROTEIN	227859
9	[]	Severe acute respiratory syndrome coronavirus 2	Replicase polyprotein 1ab	4.0	False	CHEMBL4523582	[{'accession': 'P0DTD1', 'component_descriptio...	SINGLE PROTEIN	2697049

From this, the subset of bioactivity data with IC50 as the unit was selected for further investigation. It should be noted that some compounds may have multiple bioactivity data points as reported from more than one research article owing to the fact that they may have been used as reference compounds.

assay_description	assay_type	assay_variant_accession	assay_variant_mutation	bao_endpoint	...	target_organism	target_pref_name	target_tax_id	text_value	toid	type	units	uo_units	upper_value	va
In vitro inhibitory concentration against SARS...	B	None	None	BAO_0000190	...	SARS coronavirus	SARS coronavirus 3C-like proteinase	227859	None	None	IC50	uM	UO_0000065	None	
In vitro inhibitory concentration against SARS...	B	None	None	BAO_0000190	...	SARS coronavirus	SARS coronavirus 3C-like proteinase	227859	None	None	IC50	uM	UO_0000065	None	
In vitro inhibitory concentration against SARS...	B	None	None	BAO_0000190	...	SARS coronavirus	SARS coronavirus 3C-like proteinase	227859	None	None	IC50	uM	UO_0000065	None	

5.2 Data preprocessing

After the data collection and control, Data sets were pre-processed by removing redundant and missing data. We got non-redundant SMILES of 133 experimentally validated unique compounds. Then, the standard value norm were converted from nanometer to meter via multiplying by $1*10^{-9}$ meters. Further, we also converted the IC50 into the negative logarithm of half maximal inhibitory concentration (pIC50). The pIC50 is calculated using the formula: $pIC50=-\log_{10}(IC50)$,where IC50 is the half-maximal inhibitory in Molar concentration. The RDKit was used for generating Lipinski Descriptors and molecular fingerprints from the SMILES. Data preprocessing is an important step for getting the maximum amount of descriptors, and fingerprints which helps to explore and extract the most relevant features.

	molecule_chembl_id	canonical_smiles	standard_value	pIC50
0	CHEMBL187579	Cc1noc(C)c1CN1C(=O)C(=O)c2cc(C#N)ccc21	7200.0	5.142668
1	CHEMBL188487	O=C1C(=O)N(Cc2ccc(F)cc2Cl)c2ccc(I)cc21	9400.0	5.026872
2	CHEMBL185698	O=C1C(=O)N(CC2COc3ccccc3O2)c2ccc(I)cc21	13500.0	4.869666
3	CHEMBL426082	O=C1C(=O)N(Cc2cc3ccccc3s2)c2ccccc21	13110.0	4.882397
4	CHEMBL187717	O=C1C(=O)N(Cc2cc3ccccc3s2)c2c1cccc2[N+](=O)[O-]	2000.0	5.698970
...	...	...	...	...
128	CHEMBL2146517	COC(=O)[C@@]1(C)CCCc2c1ccc1c2C(=O)C(=O)c2c(C)c...	10600.0	4.974694
129	CHEMBL187460	C[C@H]1COC2=C1C(=O)C(=O)c1c2ccc2c1CCCC2(C)C	10100.0	4.995679
130	CHEMBL363535	Cc1coc2c1C(=O)C(=O)c1c-2ccc2c(C)cccc12	11500.0	4.939302
131	CHEMBL227075	Cc1cccc2c3c(ccc12)C1=C(C(=O)C3=O)[C@@H](C)CO1	10700.0	4.970616
132	CHEMBL45830	CC(C)C1=Cc2ccc3c(c2C(=O)C1=O)CCCC3(C)C	78900.0	4.102923

133 rows × 4 columns

5.3 Features used

Molecular descriptors are quantitative and/or qualitative description of the chemical information encoded within chemical structures that are used for subsequent model building. The RDKit software was used for computing 4 sets of molecular descriptors based on Lipinski's Rule of 5. Those descriptors are molecular weight, hydrogen bond acceptors, hydrogen bond donors and a calculated octanol-water partition coefficient (logP). Lipinski's Rule of 5 was chosen because it was mainly developed to set 'drugability' of new molecular entities (NMEs) [4].The RDKit software was used as well for computing Morgan fingerprints from the molecule SMILES. Morgan fingerprints are a way to represent molecules as mathematical objects in a circular 2-D representation which provides tangible description of the sub-structural components of investigated molecules that are thus interpretable and would provide actionable information to medicinal chemists for guiding molecular structure refinement and the lead optimization process.

5.4 Data splitting

The dataset were subjected to data splitting where 80% of the entire dataset was used as the internal set and the remaining 20% served as the external set. The internal set was used for training the regression models and its ability to extrapolate to unknown compounds are simulated by testing against the external set. Furthermore, the dataset was also used for evaluating the best model performance via a fivefold cross-validation scheme after the tuning of the best model hyperparameters.

5.5 Model evaluation:

We used Mean Squared Error (MSE) .it represents the squared distance between actual and predicted values[22]. We perform squared distance to avoid the cancellation of negative terms.

$$MSE = (1/n) * \Sigma(O_i - E_i)^2$$

Where: n is the sample size (the number of observations), O_i refers to the predicted values

E_i refers to the corresponding actual values, Σ denotes the summation of the squared differences across all observations.

R-squared [22] also known as the coefficient of determination is a statistical measure that represents the goodness of fit of a regression model. The value of R-square lies between 0 to 1. Where we get R-square equals 1 when the model perfectly fits the data and there is no difference between the predicted value and actual value. However, we get R-square equals 0 when the model does not predict any variability in the model and it does not learn any relationship between the dependent and independent variables.

$R^2 = 1 - (SSR / SST)$

Where:

SSR is the sum of squared residuals, which represents the sum of the squared differences between the predicted values and the mean of the dependent variable.

SST is the total sum of squares, which represents the sum of the squared differences between the actual values and the mean of the dependent variable.

The correlation between two variables is measured using Pearson's correlation coefficient (PCC or R). In bioinformatics, the two variables are actual and predicted values. The range of PCC varies from −1 to +1. If PCC is −1, it indicates that observed and actual values are negatively correlated, 0 shows random prediction, while +1 displayed the positive correlation among them[22]. PCC is calculated using the formula:

$$r = \frac{n\sum xy - (\sum x)(\sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}}$$

The **Pearson correlation coefficient (PCC)** was chosen as it was employed in previous literature; however, PCC is primarily useful for just identifying patterns and does not provide a comprehensive measure of the model's strength. In regression model evaluation, the **coefficient of determination (R^2)** is typically preferred as it offers a more comprehensive assessment of the model's performance. Therefore, our criteria for selecting the best model was based on the R^2 score.

5.6 Learning Process

The model was constructed using the support vector regression (SVR), Random Forest (RF), Linear Regression, and Gradient Boosting Regressor algorithms. These algorithms were selected based on previous research. Two models were developed: one for predicting the pIC50 value based on the Linpski descriptors and another based on the Morgan fingerprints. The model that utilized Lipinski descriptors was evaluated as follows:

```
Model: Random Forest
R² Score: 0.3890809173671441
Mean Squared Error: 0.470568412931906
Pearson Correlation Coefficient: 0.6477179843314248

Model: Linear Regression
R² Score: 0.4072743754046829
Mean Squared Error: 0.4565546639464075
Pearson Correlation Coefficient: 0.6639429098549039

Model: Support Vector Regression
R² Score: 0.03121881656607317
Mean Squared Error: 0.746216376155933
Pearson Correlation Coefficient: 0.2487178979256972

Model: Gradient Boosting Regressor
R² Score: 0.3350097602950063
Mean Squared Error: 0.5122174288034872
Pearson Correlation Coefficient: 0.581709980352027
Best Model: Linear Regression
```

Then, we used the same algorithms on the Morgan fingerprint and we got the following result:

```
Model: Random Forest
R² Score: 0.6688494167159084
Mean Squared Error: 0.2550730675262251
Pearson Correlation Coefficient: 0.821188031226986

Model: Linear Regression
R² Score: -1.0639993968436948e+24
Mean Squared Error: 8.195594502883438e+23
Pearson Correlation Coefficient: 0.20721949827600664

Model: Support Vector Regression
R² Score: 0.648699156994148
Mean Squared Error: 0.2705940685998368
Pearson Correlation Coefficient: 0.8164919376330121

Model: Gradient Boosting Regressor
R² Score: 0.7207196895105875
Mean Squared Error: 0.21511931155228384
Pearson Correlation Coefficient: 0.8533312126819913

Best Model: GradientBoostingRegressor(random_state=42)
```

5.7 Result and Discussion

Fingerprint-based algorithms have demonstrated superior performance compared to Lipinski descriptor-based algorithms in our study. Specifically, the Gradient Boosting Regressor yielded notable results with an R2 score of 0.72 and a PCC of 0.85. To further optimize the performance of the Gradient Boosting Regressor, we conducted hyperparameter tuning by adjusting parameters such as learning_rate, n_estimators, and max_depth. Given the relatively small dataset, we employed a 5-fold cross-validation methodology. Through this process, we achieved

an enhanced performance of 0.73 by selecting the optimal hyperparameters: learning rate = 0.01, max depth = 3, and n_estimators = 300.

Subsequently, we utilized the best-performing model, the Gradient Boosting Regressor, to develop a code that accepts an input SMILE and utilizes RDKit to compute the corresponding fingerprints. With the aid of this model, we successfully predicted the pIC50 value of the input molecule. In addition, employing a threshold of pIC50 ≥ 4.5, we classified molecules as active or inactive. Consequently, we have established a regression model capable of providing reliable predictions for the pIC50 value of a given molecule. This model facilitates informed decision-making regarding molecule selection for experimental purposes, while simultaneously providing valuable insights into the activity status of the molecules.

6. Conclusion and Future Work

Our findings support the conclusion that the regression model utilizing fingerprints outperforms the model based on Lipinski descriptors in predicting the pIC50 values of input molecules represented by SMILES. This discrepancy in performance can be attributed to the inherent limitations of Lipinski descriptors, which primarily focus on oral drug-like properties and lack specificity in capturing the unique characteristics of individual molecules. On the other hand, fingerprints as the name implies, is unique and specific to each molecule and are designed to encode the distinct structural features of molecules, allowing for a more comprehensive and specific representation that enables improved prediction accuracy.

Future research should focus on the investigation of novel fingerprints and descriptors. In addition to identifying the most effective fingerprints and Lipinski descriptors, a promising direction is to develop a combined dataset that integrates the most effective of both types of descriptors. This hybrid dataset can then be employed to train a gradient boosting regression model, with the objective of determining whether the model's R^2 value can be further improved. Such endeavors would contribute to advancing the understanding and application of these techniques in the field.

References:

1- Abdulrahman, H. L., Uzairu, A., & Uba, S. (2020, September 15). Computer modeling of some anti-breast cancer compounds. Structural Chemistry, 32(2), 679–687. https://doi.org/10.1007/s11224-020-01608-7

2- Aykul, S., & Martinez-Hackert, E. (2016). Determination of half-maximal inhibitory concentration using biosensor-based protein interaction analysis. *Analytical biochemistry*, *508*, 97–103. https://doi.org/10.1016/j.ab.2016.06.025

3- Basak, Debasish & Pal, Srimanta & Patranabis, Dipak. (2007). Support Vector Regression. Neural Information Processing – Letters and Reviews. 11.

4- Benet, L. Z., Hosey, C. M., Ursu, O., & Oprea, T. I. (2016). BDDCS, the Rule of 5 and drugability. *Advanced drug delivery reviews*, *101*, 89–98. https://doi.org/10.1016/j.addr.2016.05.007

5- Breiman, L. (2001). Random Forests. *Machine Learning* **45**, 5–32 https://doi.org/10.1023/A:1010933404324

6- Cleary, E. G., Beierlein, J. M., Khanuja, N. S., McNamee, L. M., & Ledley, F. D. (2018). Contribution of NIH funding to new drug approvals 2010–2016. *Proceedings of the National Academy of Sciences of the United States of America*, *115*(10), 2329–2334. https://doi.org/10.1073/pnas.1715368115

7- Drews, J. (2000). Drug discovery: a historical perspective. *Science*, *287*(5460), 1960–1964. https://doi.org/10.1126/science.287.5460.1960

8- Elbadawi, M., Gaisford, S., & Basit, A. W. (2021). Advanced machine-learning techniques in drug discovery. *Drug Discovery Today*, *26*(3), 769–777. https://doi.org/10.1016/j.drudis.2020.12.003

9- Feng, C., Wang, H., Lu, N., Tian, C., He, H., Lu, Y., & Tu, X. (2014). Log-transformation and its implications for data analysis. *PubMed*. https://doi.org/10.3969/j.issn.1002-0829.2014.02.009

10- Hammoudi, N., Benguerba, Y., & Sobhi, W. (2019). QSAR modeling of thirty active compounds for the inhibition of the acetylcholinesterase enzyme. *Current Research in Bioinformatics*, *8*(1), 62-65. https://doi.org/10.3844/ajbsp.2019.62.65

11- Harris, C.R., Millman, K.J., van der Walt, S.J. et al. *Array programming with NumPy*. Nature 585, 357–362 (2020). DOI: 10.1038/s41586-020-2649-2. (Publisher link).

12- Heikamp, K., Hu, X., Yan, A., & Bajorath, J. (2012). Prediction of activity cliffs using support vector machines. *Journal of Chemical Information and Modeling*, *52*(9), 2354–2365. https://doi.org/10.1021/ci300306a

13- Jarantow, S. W., Pisors, E. D., & Chiu, M. L. (2023). Introduction to the Use of Linear and Nonlinear Regression Analysis in Quantitative Biological Assays. *Current protocols*, *3*(6), e801. https://doi.org/10.1002/cpz1.801

14- Khedekar S, Mhatre N & Mendhe R (2022). Prediction of pIC50 Values for the Acetylcholinesterase (AChE) using QSAR Model. International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 09 Issue: 03 | Mar 2022,

15- Kwon, S., Bae, H., Jo, J., & Yoon, S. (2019). Comprehensive ensemble in QSAR prediction for drug discovery. *BMC Bioinformatics*, *20*(1). https://doi.org/10.1186/s12859-019-3135-4

16- Lipinski, C. A., Lombardo, F., Dominy, B. W., & Feeney, P. J. (1997). Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Advanced Drug Delivery Reviews*, *23*(1–3), 3–25. https://doi.org/10.1016/s0169-409x(96)00423-1

17- Malik, A. A., Ojha, S. C., Schaduangrat, N., & Nantasenamat, C. (2021). ABCpred: a webserver for the discovery of acetyl- and butyryl-cholinesterase inhibitors. *Molecular Diversity*, *26*(1), 467–487. https://doi.org/10.1007/s11030-021-10292-6

18- McKinney, W., & others. (2010). Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference* (Vol. 445, pp. 51–56).

19- Mendez, D., Gaulton, A., Bento, A. P., Chambers, J., De Veij, M., Felix, E., Magariños, M. P., Mosquera, J. F., Mutowo, P., Nowotka, M., Gordillo-Marañón, M., Hunter, F., Junco, L., Mugumbate, G., Lopez, M. R., Atkinson, F., Bosc, N., Radoux, C. J., Segura-Cabrera, A., . . .

Leach, A. R. (2018). ChEMBL: towards direct deposition of bioassay data. *Nucleic Acids Research*, *47*(D1), D930–D940. https://doi.org/10.1093/nar/gky1075

20- Muhammad, U., Uzairu, A., & Arthur, D. E. (2018). Review on: quantitative structure activity relationship (QSAR) modeling. *Journal of Analytical & Pharmaceutical Research*, *7*(2). https://doi.org/10.15406/japlr.2018.07.00232

21- Patel, V., & Shah, M. (2022b). Artificial intelligence and machine learning in drug discovery and development. *Intelligent Medicine*, *2*(3), 134–140. https://doi.org/10.1016/j.imed.2021.10.001

22- Rajput, A., Bhamare, K. T., Thakur, A. K., & Kumar, M. (2022). Biofilm-I: a platform for predicting biofilm inhibitors using quantitative Structure—Relationship (QSAR) based regression models to curb antibiotic resistance. *Molecules*, *27*(15), 4861. https://doi.org/10.3390/molecules27154861

23- RDKit: Open-source cheminformatics. https://www.rdkit.org

24- Rodríguez-Pérez, R., & Bajorath, J. (2022). Evolution of Support Vector Machine and Regression Modeling in Chemoinformatics and Drug Discovery. *Journal of computer-aided molecular design*, *36*(5), 355–362. https://doi.org/10.1007/s10822-022-00442-9

25- Romano, J. D., & Tatonetti, N. P. (2019). Informatics and Computational Methods in Natural Product Drug Discovery: A review and Perspectives. *Frontiers in Genetics*, *10*. https://doi.org/10.3389/fgene.2019.00368

26- Sarkar, C., Das, B., Rawat, V. S., Wahlang, J. B., Nongpiur, A., Tiewsoh, I., Lyngdoh, N. M., Das, D., Bidarolli, M., & Sony, H. T. (2023). Artificial intelligence and machine learning technology driven modern drug discovery and development. *International Journal of Molecular Sciences*, *24*(3), 2026. https://doi.org/10.3390/ijms24032026

27- Soares, I. T., Rodrigues, I. C., Da Costa, P. M., & Gales, L. (2022). Antibacterial and antibiofilm properties of Self-Assembled dipeptide nanotubes. *International Journal of Molecular Sciences*, *24*(1), 328. https://doi.org/10.3390/ijms24010328

28- Srivastava, G. N., Malwe, A. S., Sharma, A., Shastri, V., Hibare, K., & Sharma, V. K. (2020). Molib: A machine learning based classification tool for the prediction of biofilm inhibitory molecules. *Genomics*, *112*(4), 2823–2832. https://doi.org/10.1016/j.ygeno.2020.03.020

29- Vamathevan, J., Clark, D., Czodrowski, P., Dunham, I., Ferrán, E. A., Lee, G., Li, B., Madabhushi, A., Shah, P. K., Spitzer, M., & Zhao, S. (2019). Applications of machine learning in drug discovery and development. *Nature Reviews Drug Discovery*, *18*(6), 463–477. https://doi.org/10.1038/s41573-019-0024-5

30- Wishart, D. S., Djoumbou-Feunang, Y., Guo, A. C., Lo, E., Marcu, A., Grant, J. R., Sajed, T., Johnson, D. L., Li, C., Sayeeda, Z., Assempour, N., Iynkkaran, I., Liu, Y., Maciejewski, A., Gale, N., Wilson, A. C. C., Chin, L., Cummings, R., Le, D., . . . Wilson, M. (2017). DrugBank 5.0: a major update to the DrugBank database for 2018. *Nucleic Acids Research*, *46*(D1), D1074–D1082. https://doi.org/10.1093/nar/gkx1037

31- Zemel, R. S., & Pitassi, T. (2000). A Gradient-Based boosting algorithm for regression problems. *Neural Information Processing Systems*, *13*, 696–702. http://papers.nips.cc/paper/1797-a-gradient-based-boosting-algorithm-for-regression-problems.pdf

32- Zhou, S., & Zhong, W. (2017). Drug Design and Discovery: Principles and applications. *Molecules*, *22*(2), 279. https://doi.org/10.3390/molecules22020279

A Comparative Study on Machine Learning-Based Models for Predicting Student Performance

Muhammad Sammy Ahmad
Department of Computer Science
FGSSR, Cairo University
dr.muhammad.badran@gmail.com

Ahmed H. Asad
Information Systems and Technology Department
FGSSR, Cairo University
ah_hamza@cu.edu.eg

Ammar Mohammed
Department of Computer Science
FGSSR, Cairo University
ammar@cu.edu.eg

Abstract

Machine learning has been used extensively for enhancing solutions for many complex problems and for finding new solutions for unsolved ones. Using machine learning in education and analyzing data that are extracted from educational environments would significantly help improving learning process. The aim of this research is to compare six machine learning-based models, namely artificial neural network, k-nearest neighbor, random forest, support vector machine, logistic regression and naïve Bayes, for predicting performance of students using their demographic data, assessments scores and virtual learning environment activities in a benchmark dataset known as Open University Learning Analytics Dataset. Our results showed that the random forest model outperformed all other machine learning models with an accuracy of 94.68%.

Keywords: Predicting Student Performance, Educational Data Mining, Learning Analytics, Machine Learning.

I. INTRODUCTION

Data Mining (DM) is a process of using statistics and machine learning (ML) techniques and methods, in addition to database systems, to extract valuable information and find useful patterns in data. It can be used for anomalies detection, clustering analysis, and discovering dependences. The power of DM emerges from its applicability on any data that come from any domain. Using DM in education lead to the emergence of an interdisciplinary field called Educational Data Mining (EDM). After nearly two decades,

EDM has reached a relatively mature level as the number of researches that are concerned with applying DM techniques and tools on data generated by student's activities in learning environments, has significantly increased recently [Bakhshinategh et al., 2018].

EDM process starts by data acquisition, either from general public repositories or directly from educational environments. Once the data are collected, the first step in EDM process is preprocessing these data, in which data are represented in a cleaned suitable format for feature selection. The second step is using one or more of the DM techniques such as classification, prediction, or clustering. The third and last step is data post-processing in which the outcome is interpreted and decision is made to apply enhancements to the educational environment [Anjum & Badugu, 2020].

One of the most famous and challenging problems that has been tackled by EDM through the previous years is Predicting Student Performance (PSP) problem. It is considered the oldest application of DM techniques in education [Asiah et al., 2019]. It aims to forecast the academic achievement of students in the future. Early prediction of students' performances help them to take actions to avoid failure. It also helps instructors to assist their students by understanding their strengths and weaknesses and warn them before they fail. Additionally, it helps educational organizations and institutions to improve their success rates and lowers failure/drop rates.

It is very challenging to accurately predict student performance because there are many factors that directly or indirectly affect the student performance. Among these factors are students previous grades and class performance, their e-learning activity, their demographics, their socioeconomic information, their environment factors such as type of their school (mixed-schools or single-gender schools, religious or secular schools public or private schools, etc…) or type of classroom (lecture, auditoria, seminar, etc…), instructors attributes, course attributes and evaluations, and students experience [Shahiri et al., 2015].

The main contribution of this research is comparing performances of six ML-based models, namely artificial neural network (ANN), k-nearest neighbor (K-NN), random forest (RF), support vector machine (SVM), logistic regression (LR), and naïve Bayes

(NB) to predict the academic performance of students using their demographic, assessments scores, and their virtual learning environment (VLE) data from the Open University Learning Analytics Dataset (OULAD) [Kuzilek et al., 2017]. The reasons for choosing OULAD specifically are: (1) it is a benchmark dataset, (2) it contains relevant features, (3) it is relatively large, (4) it is labeled, (5) it is well documented, and (6) it is clean. The aforementioned features were selected because 74% of previous researches have been focusing on demographics, students' previous grades, and eLearning activity [Abu Saa, et al, 2019].

This paper is organized as follows: section II covers a detailed literature review on the PSP problem from the aspects of benchmark evaluation datasets, performance evaluation metrics and the state-of-art approaches. Section III presents the six ML algorithms used in this comparative study, section IV introduces the experimental work that was carried out during this research, and it includes the following subsections: software tools and libraries, results, and discussion. Finally, section V provides the conclusions and suggested future work.

II. LITERATURE REVIEW

This section is divided into three subsections. First subsection presents some of the famous educational datasets. The second one shows the most frequently used evaluation metrics in EDM. The third subsection introduces that state-of-the-art in EDM.

A. Benchmark datasets

The vast majority of EDM researchers use dataset that they have collected for the purpose of carrying out their own researches, and they usually do not publish it for public use. It is worthwhile to note that there are some publically published fictional datasets that cannot be used in research to solve problems or draw conclusions, but they were created to help data science students and statisticians to practice their knowledge and acquire skills. In this section, we are going to introduce some of the important public datasets that can be used in EDM field researches.

Student Performance Data Set [Cortez & Silva, 2008] is a multivariate dataset which is available at UCI: Machine Learning Repository. It contains 649 instance and 33 attributes

for students from two Portuguese high schools. These attributes include students' demographic data, social data, grades, and features related to the school. It can be used to investigate classification and regression tasks. It contains two files one for math course and the other for Portuguese language course.

Students' Academic Performance Dataset [Amrieh et al., 2016] is collected form Kalboard 360 Learning Management System (LMS) by using experience API which is a learner activity tracker tool. This tool monitors student's activities such watching tutorial videos or reading articles. The dataset contains 480 instances and 16 features. These attributes can be classified into 3 categories which are demographic features, academic features, and behavioral features.

Open University Learning Analytics Dataset (OULAD) [Kuzilek et al., 2017] is the dataset which was used in this research. The data are distributed into seven files which contain information about unique 28785 students (15046 males and 13739 females), including information about courses, their lengths in days, registration dates, assessments and students' results in them, the materials that are available via virtual learning environment (VLE) and students' interaction with them, and students' demographic information.

B. Frequently Used Evaluation Metrics in EDM

In this section, performance measures which are usually used in EDM are introduced. Different methods have different metrics to evaluate their performance. These performance measures include accuracy, kappa statistic, receiver-operating characteristic, precision, recall, mean absolute error, and root mean square error.

1) Accuracy

It is one of the most famous used metrics in classification problems, and although it is used a lot to evaluate classifiers, but it cannot be used alone to determine its true performance. In most cases target classes are not equally distributed or the data are unbalanced, so high accuracy rate of a model can be deceivable. To find accuracy of a model equation (1) can be used:

$$Accuracy = \frac{Number\ of\ correct\ predicted\ cases}{Total\ number\ of\ cases} \quad (1)$$

2) Kappa Statistic

Kappa value measures how much the classifier is close to ground truth. It is used to compare between an observed accuracy which is mentioned in the previous section and expected accuracy which can be generated by a random classifier by chance. It can be used to evaluate a classifier and compare its performance with other classifiers on the same dataset as well. Kappa value can be evaluated using equation (2):

$$Kappa = \frac{Observed\ Acc. - Expected\ Acc.}{1 - Expected\ Acc.} \quad (2)$$

3) Receiver Operating Characteristic

It can be used with binary classifiers and multiclass classifiers. The x-axis of the plot represents the false positive rate (FPR) while the y-axis represents the true positive rate (TPR). Calculating the area under the curve (AUC) shows the performance of the classifier.

4) Precision and Recall

Precision measures the ability of the classifier to classify an instance as belong to a specific class is actually true, so it is the ratio of true positive (TP) to the sum of TP and false positive (FP). Finding precision can be achieved using equation (3):

$$precision = \frac{TP}{TP + FP} \quad (3)$$

Recall measures the ability of the classifier to truly classify instances of a certain class as true, so it is the ratio of the TP to the sum of TP and false negatives (FN). Finding recall can achieved using equation (4):

$$recall = \frac{TP}{TP + FN} \qquad (4)$$

5) F_1 Score

The general formula to measure F score for positive real β is called F-beta score which can be calculated from formula (5), and it is the harmonic mean of precision and recall:

$$F_\beta = (1 + \beta^2).\frac{precision.recall}{(\beta^2.precision) + recall} \qquad (5)$$

The default value of β is 1.0, so in this case it is called F_1 score. F_1 reaches its best value at 1, which means perfect, and its worst value is 0.

The equation of F_1 score can be written as in formula (6):

$$F_1 = \frac{2 \times precision \times recall}{precision + recall} \qquad (6)$$

6) Mean Absolute Error (MAE)

It is a metric which is used in regression problems, and it is the average of absolute values of subtracting the predicted value from the actual value. Evaluating the mean absolute error is done using equation (7):

$$MAE = \frac{\Sigma_{i=1}^{n}|y_i - x_i|}{n} \qquad (7)$$

Where, y_i is the actual value, x_i is the predicted value, and n is the number of instances or data points.

7) Root Mean Square Error (RMSE)

It is another metric which is used with regression problems, and it is used to measure the difference between the predicted values and the actual values. The advantage of using root mean square error above mean absolute error, is that the greater deviation of the predicted value from the actual value, the greater the penalty on the model due to the squaring operation. Equation (8) is used to evaluate RMSE.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \hat{x}_i)^2}{n}} \qquad (8)$$

Where, x_i is the actual value, $\hat{x}_i$ is the predicted value, and n is the number of instances or data points.

C. The State-of-Art

Researchers in [Hooda et al., 2022] used an enhanced Fully Connected Network (FCN) and compared its performance to a regular ANN. Their approach is divided into four stages which are (1) data acquisition where they gather data, (2) data orientation where they represent data in a proper file format (3) data cleaning and preprocessing and finally (4) building a machine learning-based models. They used OULAD to evaluate their FCN model. The accuracy of their model achieved 84%, and recall of 0.88, F1-score of 0.91, and precision of 0.93. The result of their regular ANN: accuracy of 74%, recall of 0.72, F1-score of 0.76 and precision: 0.81. They concluded that their FCN performs better than the regular ANN.

The Covid-19 pandemic forced many educational organizations to depend heavily on online learning platforms. This increased the amount of data that can be used in EDM and motivated more researchers to try different methods. For example, in [Ali & Perumal, 2022], they predicted students' performances using Convolutional Neural Network (CNN) to extract features, Minimum Redundancy Maximum Relevance (mRMR) method to filter the extracted features, Stochastic Gradient Descent (SGD) to measure and update features' weights, and finally they used Linear Discriminant Analysis (LDA) to classify students' grades into three classes, low, medium, and high grades. The accuracy of their model reached 96.5%.

In another research [Brahim, 2022], they monitored students' interaction during online lab to gather data about 86 features such as the number of keystrokes, type of activity engaged by the student, and time spent by the student in these activities, etc. They used five different classifiers to predict students' performance. They evaluated their model

under three different scenarios, which are (1) training on all lab session except one and test on that one, (2) splitting data to 80% training and 20% for testing, and (3) five cross-validation. The highest classification accuracy of 97.4% was achieved by the RF classifier.

Some researchers enhance algorithms to achieve better prediction results. Such as in [Turabieh et al., 2021], they avoided being trapped in local optimum and rapid convergence problem by tweaking Harris Hawks Optimization (HHO) algorithm via dynamic regulation of population diversity using k-NN algorithm as a clustering approach. Their approach aimed to select the most important features that can be used to solve student performance prediction problem. They used got their dataset from UCI machine learning repository and different classifiers to evaluate their prediction system. The results showed that using both the improved HHO algorithm and Layered Recurrent Neural Network (LRNN) together outperform other classifiers with accuracy reaches 92%.

In [Li & Liu, 2021], they collected data starting from 2007 to 2019 from multidisciplinary universities and used both linear regression and deep learning to predict student performance in Higher Education. They fed data from years 2007 to 2016 to train their model during training phase and used the rest of the data for testing phase. In their model, feed forward and Backpropagation algorithms update the number of hidden layers and nodes automatically. They evaluated their model using MAE and RMSE. The achieved results were 0.593 and 0.785 respectively.

Another research [Heise et al., 2020] was conducted to investigate a possible correlation between the examinations scores of human dissection course and the weekly table quizzes to predict the learning outcomes. They analyzed data that were generated during 5 years, namely: 2012, 2013, 2015, 2016, and 2017. They found that, in 2012 when there was no oral component in the quizzes, there was no correlation between the examination scores of the course and the performance of the students in quizzes. They attributed this to the absence of active learning environment as there was no interaction between faculty staff and students during the table quizzes which lasted only about 10 minutes, but in years 2013, 2015, 2016, and 2017, there was a positive correlation between the examination scores and the performance of students in the table quizzes,

when the oral component was included in the quizzes converting them to a conversation-based assessment for about 20-25 minutes from which the students had immediate feedback on their current level and enhanced the communication between the students and the faculty staff.

In [Moreno-Marcos et al., 2020], they analyzed several factors that has an impact on predicting students' performances. These factors, include but not limited to, students' grades, exercises-related variables, course-related variables such its duration, types of assignments, exam questions format, clickstream data, and forum variables. They found that exercise-related variables are the most important predictors, while they found that forum variables are useless. Clickstream data are acceptable as predictors and can be used when exercise variables are not available, but they do not add any prediction power, if exercise variables exist. They also found that coding questions are harder to predict than multiple-choice questions, while the final course grade which reflects the average knowledge gained during a long relatively long time span is easier to predict than assignments grade which reflects actual knowledge acquired by the student at a specific moment.

In [Lau et al., 2019], they investigated prediction of performance of university undergraduate students. Their dataset contained little less than 1000 students (810 males and 175 females) from University Q in China. Initially, they determined the factors which were expected to affect students' performances using conventional statistical methods and listed 11 input variables. Then, they built an ANN that was composed of two hidden layers and an output layer and used Levenberg-Marquardt algorithm for Backpropagation. They assessed their model using different methods and its overall accuracy was 84.8%. It is worthwhile to note that high number of false negatives were obtained and their model inefficiently classified students based on their gender because their dataset was unbalanced.

III. COMPARATIVE STUDY

A. Artificial Neural Network (ANN)

The nodes of ANNs are organized in three different types of layers, namely the input layer, hidden layer(s), and output layer: In complex problems, the number of nodes in the input layer can be very high which may lead to the curse of dimensionality problem. After the input layer, there is one or more hidden layers. The final layer is the output layer. The number of nodes in the output layer depends on the number of classes of the classification problem. The connections between nodes are weighted, and these weights are continuously updated by Backpropagation [Umair & Sharif, 2018].

In this research, the ANN consisted of 5 layers. The input layer contained 31 nodes, each hidden layer contained 128 nodes, and the output layer contained only 3 nodes, one for each class. The activation function for the inner nodes was the Rectified Linear Unit (ReLU). It can be defined by the equation (9):

$$R(x) = max(0, x) \qquad (9)$$

Softmax function was used as the activation function of the output layer to calculate the probability of each class. Softmax function can be defined by the equation (10):

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \qquad (10)$$

Where σ is the softmax function, $\vec{z}$ is the input vector, e^{z_i} is the standard exponential of the input vector, and e^{z_j} is the standard exponential of the output vector.

The number of epochs was 50 epochs and the batch size was 64, and Adam optimizer [Kingma & Ba, 2014] was used.

B. K-Nearest Neighbor (k-NN)

This algorithm can be used for both classification and regression. In case of classification, the algorithm measures the distance between the new data point and k original data points in the dataset, and then assign the new point to the class which has the highest number of closest points to it. That's why it is called k-Nearest Neighbor, or k-

NN, where k is a number defined by the user depending on the data that are used. The value of k is usually odd to avoid equal votes.

If the data contain noise, large k reduces the effect of noise which severely affect the performance of the algorithm. The distance can be measured by Euclidian distance for continuous variables or Manhattan distance for discrete variables, or any other measure of distance can be used. If the features of the data have different scales, normalizing features will significantly improve the performance of the algorithm in classification. If the classes are skewed, then more present classes would dominate over the less present one. This problem can solved by assigning different weights to classes or by using self-organizing maps (SOM).

In our research, the value of k was 5 and the distance metric was the Euclidian distance metric by setting the value of metric to Minkowski and the power p to 2.

C. Random Forest (RF)

Decision Tree (DT) maps from observations to target values and can be used in classification and regression problems. There are many algorithms for DTs such as ID3 (Iterative Dichotomiser 3) and C4.5 which is a successor for ID3 [Quinlan, 1986; Quinlan 2014]. A single DT is liable to suffer from overfitting on the training data, so bootstrap aggregated trees, or bagged trees, are used as an ensemble method for building multiple DTs as a one of the solutions for the overfitting problem. Number of bagged trees are called random forest. Each tree in the random forest gives its vote, the vote which is chosen by the largest number of trees is the final result from the forest [Breiman, 2001].

The number of decision tress in our random forest was 100 trees and the chosen criterion for optimum splitting of data was entropy. Entropy $H(x)$ is a measure of information disorder. It can be calculated from equation (11):

$$H(x) = -\sum_{i=1}^{n} P(x_i) \log_2 P(x_i) \qquad (11)$$

Where $P(x_i)$ is the probability of a category and i is the index that indicates the number of available categories.

D. Support Vector Machine (SVM)

It is a classification method which chooses the best decision boundary that is as far as possible from linearly separable classes. The closest points to the decision boundary is called support vectors, and the largest the distance between the support vectors is called the functional margin. The classification is called the large margin classification when the classifier tries to find the widest separation possible between support vectors, while it is called hard margin classification when data points must be strictly classified into different groups that are separated as far as possible. This makes hard margin classification sensitive to outliers, so soft margin classification can be used to allow some points to not be as far as possible or even in the incorrect group, and this would make a better generalized model.

It can do nonlinear classifications by mapping inputs into high-dimensional space using maximum-margin algorithm. A simple modification to this algorithm is known as kernel trick which is replacing dot product with feature vectors by one nonlinear kernel functions. SVM Classifiers are also sensitive to feature scaling, so normalizing features gives better results. SVM can be used for categorizing text and hypertext, image classification, handwriting recognition, and protein classification.

In our research, we used SVM with the default Radial Basis Function (RBF) kernel function, which can be formulated in the following formula (12):

$$K(x, x') = e^{-\gamma \|x - x'\|^2} \qquad (12)$$

Where γ is a scalar that can be manually set, but it should be more than zero and indicates the effect that each training example, and $\|x - x'\|^2$ is the squared Euclidean distance between two points.

E. Logistic Regression (LR)

It is a classification method that gives the probability of a data point to belong to specific class. It can be used to classify binary classes, where a threshold (usually 0.5) is set to distinguish the positive class, $\hat{y} = 1$, from the negative class, $\hat{y} = 0$. After

computing the weighed sum of the input features as in linear regression, it is used as an input in a logistic function, or sigmoid function (σ). The sigmoid function is shown below in equation (13):

$$\sigma(t) = \frac{1}{1 + e^{-t}} \qquad (13)$$

The estimated probability ($\hat{p}$) can be calculated from equation (14):

$$\hat{p} = \sigma(X^T . \theta) \qquad (14)$$

Where X^T is the transpose of the input features and θ is the weight vector.

A cost function must be used to adjust the weight vector (θ). The cost function of one instance is shown below in equation (15):

$$c(\theta) = \begin{cases} -log(\hat{p}) & if\ y = 1 \\ -log(1 - \hat{p}) & if\ y = 0 \end{cases} \qquad (15)$$

The cost function for the whole training set is shown in equation (16) below:

$$j(\theta) = -\frac{1}{m} \Sigma_{i=1}^{m} \left[y^{(i)}\, log\left(\hat{p}^{(i)}\right) + \left(1 - y^{(i)}\right) log\left(1 - \hat{p}^{(i)}\right) \right] \qquad (16)$$

The cost function is convex, so gradient descent can be used to find the global minimum and find the optimum weight vector.

F. Naïve Bayes (NB)

It is a classifier that uses Bayesian theorem in statistics to find the probability of a data point to belong to certain class. It is called naïve because it assumes complete independence between features of dataset that is why it is also called simple Bayes. Assuming the independence between features is not accurate, but it helps in mitigate the problem of curse of dimensionality. Naïve Bayes may not produce accurate estimation of class probabilities, but it can contribute in solving complex real-life problems. The Bayesian Probability equation is expressed in equation (17):

$$Posterior = \frac{prior \times likelihood}{evidence} \qquad (17)$$

A class's prior can be calculated by two methods. The first one is assuming the all classes have equal probability, while the second methods is estimating probability of each class from the training set. Event model of a classifier is the assumption on distribution of features. If the features are continuous, then Gaussian distribution (or normal distribution) is frequently used, while in case of discrete data, then Bernoulli and multinominal naïve Bayes are frequently used. In our research, we used Gaussian Naïve Bayes algorithm.

IV. EXPERIMENTAL WORK

A. Preparing Data

Only the "studentinfo", the "studentAssessment" and "studentVLE" tables from OULAD were used in this study to predict student academic performance. Indices of multiple deprivation (imd) of North Region and Ireland are not compatible with the system which is used in other regions of the UK, so observations from these regions were excluded from this study. The dependent vector, which is the "final_result" column in "studentinfo" table, contains four values. They are "Withdrawn," "Fail," "Pass," and "Distinction." Students who withdrew and did not finish their modules were excluded from this study due to their missing and incomplete data. Finally, we ended up with three categories which are "Distinction," "Pass," and "Fail." Synthetic Minority Oversampling Technique (SMOTE) was used to overcome the problem of imbalanced instances of the "Fail" and "Distinction" classes with respect to "Pass" class in OULAD dataset because SMOTE is considered one of the best oversampling techniques as it uses the existing instance to create similar synthetic instances.

B. Models Evaluation

1) Confusion Matrix (CM)

In confusion matrix, the rows represent the predicted classes while the columns represent the actual classes. The cells are of the CM are the true positives, false positives, true negatives, and false negatives of each class.

True positives are the test cases in which the actual class matches the predicted class. The values of true positives can be found easily on the diagonal of the CM. False positives are the test cases in which the actual classes mismatches the predicted class. The values of

false positives are the values in the column of each class except the true positives values which are described above.

True negatives are test cases which are correctly classified as not being members of a certain class. The values of true negatives for each class can be found by getting the sum of all columns and rows in the CM except the column and row of that class. False negatives are the test cases which are incorrectly classified as not being members of a certain class. The values of false negatives are the values in the row of each class except the true positive values which are described above.

Position of true positives, false positive, true negatives and false negatives in a CM are shown in Table I.

TABLE I. POSITIONS TURE POSITIVES, FALSE POSITIVES, TRUE NEGATIVES, AND FALSE NEGATIVES IN A CM

		Predicted Classes		
		Distinction [0]	**Fail [1]**	**Pass [2]**
Actual Classes	**Distinction [0]**	$TP_{Distinction}$ TN_{Pass} TN_{Fail}	FP_{Fail} $FN_{distinction}$ TN_{Pass}	FP_{Pass} $FN_{distinction}$ TN_{Fail}
	Fail [1]	$FP_{distinction}$ TN_{Pass} FN_{Fail}	TP_{Fail} $TN_{distinction}$ TN_{Pass}	FP_{Pass} $TN_{distinction}$ FN_{Fail}
	Pass [2]	$FP_{distinction}$ FN_{Pass} TN_{Fail}	FP_{Fail} $TN_{distinction}$ FN_{Pass}	TP_{Pass} $TN_{distinction}$ TN_{Fail}

Confusion matrices of all models are shown in tables II, III, IV, V, VI, and VII respectively.

TABLE II. CM OF THE ANN MODEL

		Predicted Classes		
	Classes	Dist.	Fail	Pass
Actual Classes	Dist.	94	0	3
	Fail	0	84	11
	Pass	4	9	77

TABLE III. CM OF THE RF MODEL

		Predicted Classes		
	Classes	Dist.	Fail	Pass
Actual Classes	Dist.	87	0	10
	Fail	4	78	13
	Pass	11	7	72

TABLE IV. CM OF THE K-NN MODEL

		Predicted Classes		
	Classes	Dist.	Fail	Pass
Actual Classes	Dist.	94	0	3
	Fail	0	84	11
	Pass	4	9	77

TABLE V. CM OF THE LR MODEL

		Predicted Classes		
	Classes	Dist.	Fail	Pass
Actual Classes	Dist.	92	0	5
	Fail	0	79	16
	Pass	3	10	77

TABLE VI. CM OF THE NB MODEL

		Predicted Classes		
	Classes	Dist.	Fail	Pass
Actual Classes	Dist.	90	0	7
	Fail	13	94	18
	Pass	19	5	66

TABLE VII. CM OF THE SVM MODEL

		Predicted Classes		
	Classes	Dist.	Fail	Pass
Actual Classes	Dist.	85	0	12
	Fail	2	75	18
	Pass	2	4	84

2) Accuracy

Table VIII summarizes the accuracy of all models on demographic, assessment scores, and VLE activities.

TABLE VIII. COMPARING ACCURACIES OF ALL ML MODELS

Model	RF	ANN	SVM	k-NN	LR	NB
Accuracy	**94.68%**	90.43%	86.52%	84.04%	87.94%	78.01%

As shown from the previous table, the accuracy of RF model was higher than other models. The performance of k-NN, LR and SVM are close to each other. Finally, the performance of Naïve Bayes was very poor as it had the lowest prediction accuracy.

3) Precision, Recall and F_1 Score

Tables VIII, IX, X, XI, XII, and XIII respectively compare precision, recall, F_1 score, and support of the six ML models for the three classes, namely distinct, fail and pass. In most cases, RF has the best precision, recall, and F_1 score in the three classes among the other ML models.

TABLE IX. ANN MODEL

Metric	Dist.	Fail	Pass
Precision	0.959	0.903	0.846
Recall	0.969	0.884	0.856
F_1 score	0.964	0.894	0.851
Support	97	95	90

TABLE X. CM OF THE RF MODEL

Metric	Dist.	Fail	Pass
Precision	0.969	0.946	0.931
Recall	0.989	0.936	0.911
F_1 score	0.979	0.941	0.921
Support	97	95	90

TABLE XI. CM OF THE K-NN MODEL

Metric	Dist.	Fail	Pass
Precision	0.870	0.917	0.757
Recall	0.897	0.821	0.800
F_1 score	0.883	0.867	0.778
Support	97	95	90

TABLE XII. CM OF THE LR MODEL

Metric	Dist.	Fail	Pass
Precision	0.968	0.887	0.785
Recall	0.948	0.831	0.855
F_1 score	0.958	0.858	0.819
Support	97	95	90

TABLE XIII. CM OF THE NB MODEL

Metric	Dist.	Fail	Pass
Precision	0.737	0.927	0.725
Recall	0.927	0.673	0.733
F_1 score	0.821	0.780	0.729
Support	97	95	90

TABLE XIV. CM OF THE SVM MODEL

Metric	Dist.	Fail	Pass
Precision	0.955	0.949	0.736
Recall	0.876	0.789	0.933
F_1 score	0.913	0.862	0.823
Support	97	95	90

The values in Tables IX, X, XI, XII, XII, and XIV are represented graphically in figure 1 for easier comparison between them visually.

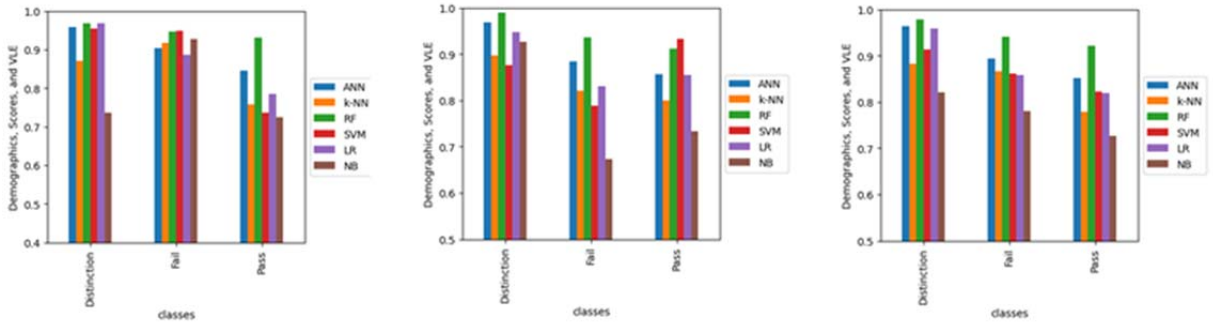

Fig. 1. Precision (left), recall (middle) and F_1 score (right) of the three classes for each model

4) Recevier Operating Characteristic (ROC) Curve

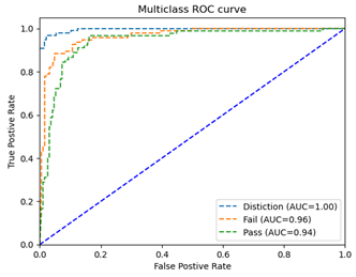

Fig. 2. ROC curve for ANN model

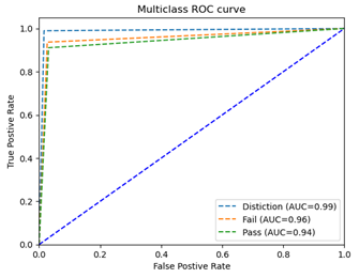

Fig. 3. ROC curve for RF model

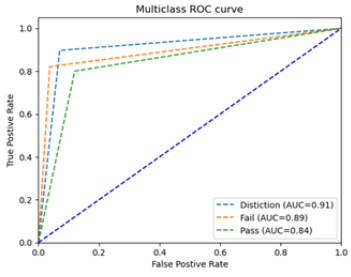

Fig. 4. ROC curve for k-NN model

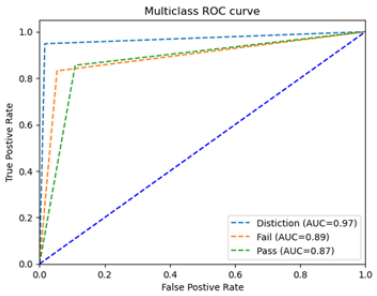

Fig. 5. ROC curve for LR model

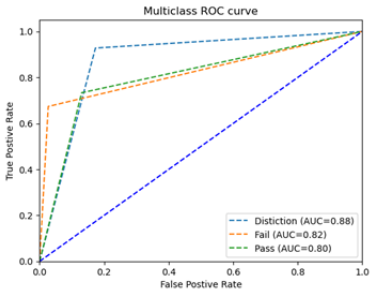

Fig. 6. ROC curve for NB model

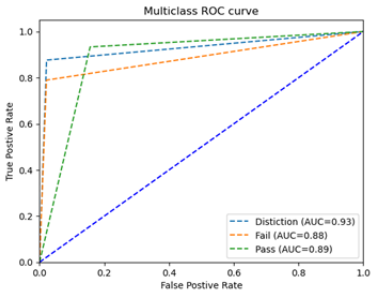

Fig. 7. ROC curve for SVM model

ROC curves of ANN and RF had the highest AUCs, where ANN got 1.00, 0.96, and 0.94 for distinction, fail, and pass respectively while RF got 0.99, 0.96, 0.94 for the three classes as the in the same aforementioned order.

V. CONCLUSIONS AND FUTURE WORK

Performance of six different ML models were compared for predicting academic performance of students of the Open University in the United Kingdom. RF accuracy of prediction reached 94.68% which is the highest compared to accuracies that were obtained from other researches that dealt with similar problems. The RF models also outperformed all other models accuracies and other evaluation metrics such as precision, recall, and F_1 score. Other researches can use deep learning or different variants of ANNs such as Convolution Neural Network (CNN) or Recurrent Neural Networks (RNN), or mine data of students' activities on the virtual learning environment (VLE) to discover patterns of their learning behavior and match these patters with their performance.

REFERENCES

Abu Saa, et al. (2019) Factors Affecting Students' Performance in Higher Education: A Systematic Review of Predictive Data Mining Techniques.

Ali S, R., Perumal, S. (2022). Multi-class LDA classifier and CNN feature extraction for student performance analysis during Covid-19 pandemic, International Journal of Nonlinear Analysis and Applications, 13(1), pp. 1329-1339.

Amrieh, E., Hamtini, T., & Aljarah, I..(2016). Mining educational data to predict student's academic performance using ensemble methods. International Journal of Database Theory and Application 9(8), 119-136.

Anjum N., Badugu S. (2020). A Study of Different Techniques in Educational Data Mining. In: Satapathy S., Raju K., Shyamala K., Krishna D., Favorskaya M. (eds) Advances in Decision Sciences, Image Processing, Security and Computer Vision. Learning and Analytics in Intelligent Systems, vol 4. Springer, Cham.

Asiah, M., Zulkarnaen, K. N., Safaai, D., Hafzan, M. Y. N. N., Saberi, M. M., & Syuhaida, S. S. (2019). A review on predictive modeling technique for student academic performance monitoring. In proceedings of materials science, engineering and chemistry (MATEC) web of conferences, 255 (03004).

Bakhshinategh, B., Zaiane, O.R., ElAtia, S. et al. (2018). Educational data mining applications and tasks: A survey of the last 10 years. Educ Inf Technol 23, 537–553.

Brahim, G.B. (2022) Predicting Student Performance from Online Engagement Activities Using Novel Statistical Features. *Arab J Sci Eng*.

Breiman, L. (2001) Random Forests. Machine Learning 45(1), 5–32.

Cortez, P. & Silva, A. (2008) Using Data Mining to Predict Secondary School Student Performance. In A. Brito and J. Teixeira Eds., Proceedings of 5th FUture BUsiness TEChnology Conference (FUBUTEC 2008), pp. 5-12, Porto, Portugal, April, 2008,

Heise, N. et al (2020). Table Quizzes as an Assessment Tool in the Gross Anatomy Laboratory. Journal of medical education and curricular development vol. 7.

Hooda, M, et al, (2022) Integrating LA and EDM for Improving Students Success in Higher Education Using FCN Algorithm. Mathematical Problems in Engineering.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kuzilek J., Hlosta M., Zdrahal Z. (2017). Open University Learning Analytics Dataset. Sci. Data 4:170171 doi: 10.1038/sdata.2017.171.

Lau, E.T., Sun, L. & Yang, Q. (2019). Modelling, prediction and classification of student academic performance using artificial neural networks. SN Appl. Sci. 1, 982.

Li, S. & Liu, T. (2021). Performance Prediction for Higher Education Students Using Deep Learning. https://doi.org/10.1155/2021/9958203

Moreno-Marcos, P., et al. (2020). Analysis of the Factors Influencing Learners' Performance Prediction with Learning Analytics. IEEE Access 8, pp. 5264-5282.

Quinlan, J. R. (1986). Induction of decision trees. Machine learning, 1(1), 81-106.

Quinlan, J. R. (2014) C4.5: programs for machine learning. Elsevier.

Shahiri, A. & Husain, W. & Abdul Rashid, N. (2015). A Review on Predicting Student's Performance Using Data Mining Techniques. Procedia Computer Science. 72. 414-422.

Turabieh, H., Azwari, S.A., Rokaya, M. et al. (2021). Enhanced Harris Hawks optimization as a feature selection for the prediction of student performance. Computing 103, 1417–1438.

Umair, S., & Sharif, M. M. (2018). Predicting students grades using artificial neural networks and support vector machine. In Encyclopedia of Information Science and Technology, Fourth Edition (pp. 5169-5182). IGI Global.

Traffic Flow Prediction with Big Data and Machine Learning Techniques

Sayed A. Sayed*, Yasser Abdel-Hamid, and Hesham Ahmed Hefny
Computer Science Department
Faculty of Graduate Studies for Statistical Research
Cairo University, Giza, Egypt
*se.sayedahmed@gmail.com , yasserabdelhamid@gmail.com , hehefny@cu.edu.eg

***Abstract*— The rise of Artificial Intelligence (AI) and Big Data in many industries, including transportation, has inspired new ideas and solutions to complicated problems like traffic congestion, which affects our quality of life. These methods are called Intelligent Transportation Systems (ITS). ITS addresses numerous traffic challenges, including congestion. Additionally, various smart city applications that improve transit and mobility use it. Traffic forecasting is essential in transportation. In addition to route planning and traffic restrictions, it can considerably impact road construction and project design. Therefore, it must be assessed and predicted precisely. Thus, traffic prediction must be accurate and efficient. To increase traffic flow prediction accuracy, new models and frameworks have been rapidly created together with AI and Big Data methods. This study's main goals are: First, to analyze the most frequent traffic prediction machine learning methods. Second, it describes traffic forecast data types. Then, it discusses BD's traffic forecast potential. It concludes by discussing machine learning and big data difficulties and future improvements in traffic prediction.**

Keywords— AI; ITS; Traffic Congestion; Machine Learning; Traffic Prediction

I. INTRODUCTION

As cities have become more crowded and congested over the last few decades, there has been a greater need for the development of ITS-based solutions for accurate traffic prediction and mobility control (Nellore K and Hancke., 2016). ITS is a new way to provide transportation that uses advanced data communication technologies. It does this by combining computers, information technology, communications technologies, and other technologies and using them in the transportation business. The goal of this process is to make a system that works for people, cars, and roads (Patel P, et al., 2019). ITS can be used to build a complete, accurate, real-time, and useful transportation control system (An S, et al., 2011). Besides that, it could cut down on dangers, high accident rates, traffic jams, carbon emissions, and air pollution while also making things safer and more reliable, speeding up travel, easing traffic, and making passengers happier (Qureshi K and Abdullah, 2013).

Accurately predicting traffic flow is an important part of ITS because it helps all the stakeholders involved (passengers, traffic managers, lawmakers, and road users) use transport networks more safely and smartly (Chen C, et al, 2020) (Sun P and Tao, 2020). The quality of the traffic data is what makes these systems work. Only then ITS will be useful. The World Health Organization's (WHO) 2018 report on the universal status of road safety says that road traffic deaths are still going up, with 1.35 million deaths recorded in 2016. This means that studying traffic forecasting is a good way to cut down on traffic and make travel safer and cheaper (Makaba T and Paul, 2020) (World Health Organization, 2018).

In the past, predicting traffic flow relied on parametric models like time series analysis that were based on data from the past. In time series, a set of readings x that were taken at a certain time t are recorded. The goal is to find patterns in past traffic data that show how traffic changed over time and then use these patterns to make predictions. The Kalman Filtering method for time-series analysis (Bengio Y, 2009) was another model for mobile random problems that could solve regression problems and reduce variance to get the best results. Also, the Auto-Regressive Integrated Moving Average (ARIMA) model is a standard and well-known way to guess how traffic will move in the short run (Van.D et al.,1996). There were many changes made to the ARIMA model, and the results showed better performance (Lee S, & Fambro D., 1999 - Williams B., 2001 - Williams B. and Hoel L, 2003 – Chen K,et al., 2020).

Nonparametric models like the Random Forest (RF) Algorithm, the Bayesian Algorithm (BA) method, K-Nearest Neighbor (KNN), Principal Component Analysis (PCA), and Support Vector Algorithms (Bengio Y, 2009) have recently been used to predict traffic flow because it is random and doesn't follow a straight line. Neural networks were also often used to predict how traffic would move (Kashyap A et al., 2022).

There are many areas that have used Big Data (BD) extensively. One of them is the transportation industry. Using a number of different data sources, it is possible to accurately predict and estimate traffic flows, which improves the operation as a whole. Transport is a great example of an area where BD has been used too much. As the Internet of Things (IoT), Cyber-Physical Systems (CPS), and smart cities grew, they made it easier to collect huge amounts of data from things like security cameras, mobile phones, and static sensors. From Trillion bytes to Petabytes (W. Jiang and J., 2022), the amount of data has grown. Several applications consider predicting traffic flow as a major challenge, and that's what this study will focus on. It is sometimes impossible to get traffic

information (like volume, speed, trip time, etc.) without changing the raw data, but it is possible to get other information about traffic conditions.

The main goals of this research are to analyze the most widely used machine learning approaches for traffic flow prediction. In addition, it presents the potential BD role in traffic prediction. Finally, it discusses the challenges facing and potential future developments in machine learning and big data concerning traffic prediction.

Here's how the remainder of this survey is put together: Section 2 talks about how the traffic prediction problem, machine learning, and big data work from a theoretical point of view. Section 3 explains how the survey was done and gives a review of the research on how machine learning and big data were utilized to predict traffic flow. In Section 4, we talk about the problems that already exist in this survey's topic. Section 5 is the last part of the paper which concludes it.

II. BACKGROUND

Many high-resolution traffic data from ITS can be used to estimate traffic flow (J. Zhang et al., 2011). Predicting traffic flow is a time series problem that uses data from past observation stations to estimate future flow counts. Traffic modeling, operation, and management involve anticipating traffic flow. Real-time traffic predictions can help road travelers save money and make better judgments. It can help traffic authorities reduce congestion by improving traffic control. AI branch machine learning has grown rapidly in recent years (Chowdary G, 2021). These approaches accurately forecast traffic. Many excellent traffic data collection systems and Big Data technology for storing and analyzing big data sets have enabled many traffic volume prediction methods.

1. Machine Learning

Machine Learning (ML) approaches are statistical models used to classify and predict outcomes based on input data (Chowdary G, 2021). ML is a branch of AI that concentrates on creating prediction algorithms by uncovering patterns in large datasets, without being tailored for a specific task (Singh G, 2018). ML models are categorized into three groups based on the learning methodologies they utilize: supervised learning, unsupervised learning, and reinforced learning (RL), as depicted in Figure 1.

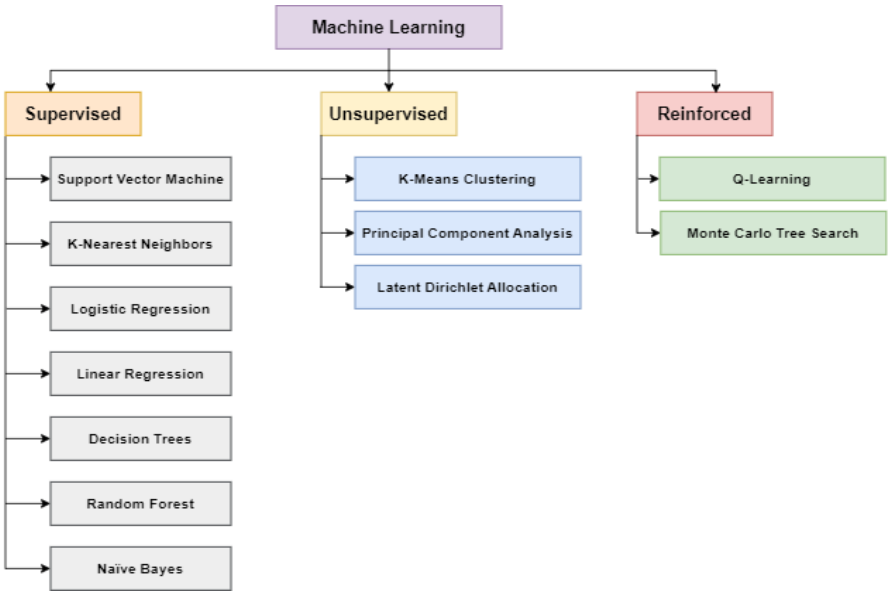

Fig. 1. *Machine Learning Categories.*

1.1 Supervised Learning

In supervised learning tasks, the model is provided with a labeled dataset consisting of feature vectors and their corresponding anticipated output labels. The goal of these models is to establish an inference function that accurately associates feature vectors with corresponding output labels. Upon the completion of ML model training, it is capable of generating predictions using novel data. Supervised learning methods (Chowdary G, 2021) can be used to provide predictions that are either continuous or discrete. Supervised learning algorithms, such as Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Logistic Regression, Linear Regression, Decision Trees (DT), Random Forests (RF), and Naive Bayes, are supervised learning approaches examples (Singh G, 2018).

A. Support Vector Machine

SVM is a deterministic linear classifier that focuses on classification challenges. SVM is considered the best ML algorithm. The core of SVM is margin calculation. This method depicts each data point as a point in an n-dimensional space, where n is the number of features, and each feature is a coordinate. This method analyzes vectorized data to create a hyperplane that separates the classes (Dey

A, 2016). Next, several margins are drawn between classes, and a hyperplane is created to reduce mean-squared error and maximize margin-class distance (Dhall D and Juneja M, 2020). After finding an optimal separating hyperplane for linearly separable data, support vector points are the data points on its boundary. A linear combination of these locations is then provided as the solution. We ignore the remaining data (Kotsiantis S and Pintelas P, 2007). Thus, the number of features in the training data does not affect SVM model complexity. SVMs excel at learning problems with many features compared to training examples.

Even though the SVM has a large margin to assess several hyperplanes, misclassified instances may prevent it from finding one that divides the data. A soft margin that allows misclassifications of certain training samples may help (Veropoulos K and Cristianini N, 1999). Only binary classification issues can be solved by SVM. Thus, multi-class problems must be converted into binary classification problems. Analyzing categorical data is tough, although scaling can improve results (Kotsiantis S and Pintelas P, 2007).

B. K-Nearest Neighbors

KNN is a nonparametric classification approach that makes no dataset assumptions. Its efficiency and simplicity are admired. KNN method predicts unlabeled data class using a labeled training dataset. To classify new inputs, KNN is used on clustered datasets. KNN is useful when data is unfamiliar (Taunk, K and Swetapadma A, 2019). KNN uses a range of variable values, usually 0–1, to find the closest training data points. KNN uses Manhattan, Euclidean, Minkowski, and Hamming distances. Nearest neighbors are calculated using Euclidean distance for continuous data. Categorical data use the Hamming distance function (Obulesu O et al., 2018).

Choosing the K value is the hardest part of the KNN algorithm because it affects how well and accurately the algorithm works. When predicting a class label, small K values cause noise, while large K values may cause too much fitting likelihood. It also takes longer to compute and slows down the speed of execution. The K value is estimated according to equation (1):

$$K = n\hat{}(1/2) \qquad (1)$$

Where n represents the size of the dataset.

The training data will undergo cross-validation using different K values to optimize the test results. The optimal value for test results will be determined based on the optimal precision (Obulesu O et al., 2018).

The KNN method is simple and easy to utilize. This classification method is adaptable and ideal for multi-mode classes. The KNN approach for classifying unfamiliar data is expensive. Distance between the k-nearest neighbors must be calculated. As training set size increases, algorithm computations become more difficult. Disruptive or insignificant features reduce precision. KNN keeps all training data without generalizing it. Thus, more data dimensions reduce location accuracy. KNN learns slowly since it calculates k neighbor distances (Ray S., 2019).

C. Logistic Regression

Logistic regression is a supervised learning approach for discriminating among three or more classes (Dey A., 2016). It quantifies the probability of an event occurrence using binary values (0 and 1) and is dependent on the input variables, resulting in a binomial conclusion. An example of a binomial outcome in Logistic Regression is the prediction of whether an email is classified as spam or not. Furthermore, Logistic Regression has the capability to generate ordinal outcomes, such as assigning a rating to a product on a scale of 1 to 5. Logistic Regression focuses on predicting categorical target variables (Ray S., 2019).

Logistic Regression has many advantages including its simplicity, speed, and efficiency in training and regularization. It does not require the input characteristics to be scaled. In addition, both data noise and numerous correlations have no effect when using this technique. On the other hand, non-linear issues are not ideal to logistic regression due to its linear decision surface, susceptibility to overfitting, and the necessity of complete knowledge about all independent variables (Ray S., 2019).

D. Linear Regression

A Linear Regression is a kind of supervised learning techniques in which the output value is calculated using the input value and the employed labeled datasets. Continuous variables can be simulated and anticipated using Linear Regression. It also seeks to fit data to a straight hyperplane if the relationship between variables in the data set is linear (Ray S., 2019). It can be computed using equation (2) (Obulesu O et al., 2018):

$$F(x) = mx + b + e \qquad (2)$$

Where F is the dependent variable (x) and x is the independent variable. The line has a slope of m, a y-intercept of b, and an error term of e. Depending on the previous information, a Linear Regression is a simple technique. It is also clear that a linear connection between dependent and independent variables is the most effective. On the other hand, linear regression can only forecast the outcomes. It has a problem with outliers and does not work well with non-linear data. In addition, data must be independent (Obulesu O et al., 2018).

E. Decision Trees

Popular data mining approaches include classifier creation (Kumar R and Verma R., 2012). Classification algorithms can process massive data mining data. The training set and new data can be classified using this strategy. It can also infer categorical class names

(Nikam S., 2015). DTs can perform numerous tasks, including pattern recognition and image processing (G Stein et al., 2005). DT compares numerical attributes to threshold values in various fundamental tests. Tests are ordered and coordinated (I. S. Damanik et al ., 2019). Data mining often classifies using DT (S. S. Gavankar and S. D. Sawarkar., 2017). Every tree has nodes and branches. Each node is a group characteristic that needs to be categorized, and each branch is a candidate value (Mahesh B., 2020). DT is a supervised learning method. A training model that uses learnt decision rules from training data to forecast target variable class or value is the goal (Charbuty B and Abdulazeez A, 2021). Table 1 lists the pros and cons of DT (Y. Zhao and Y. Zhang., 2008 - K. Mittal et al., 2017 - Priyanka and D. Kumar, 2020).

TABLE I. DT BENEFITS AND DRAWBACKS

Advantages	Disadvantages
1- Straightforward. 2- Quickly changed into a set of rules for making things. 3- Can sort both categorical and numerical results but can only create categorical attributes. 4- There are no preconceived ideas about how true the results are.	1-The best way to make decisions can be messed up, which can lead to bad decisions. 2-There are many levels to the decision tree, which makes it hard to understand. 3-As the number of training samples goes up, the calculations of the decision tree may get more complicated.

F. Random Forest

RF uses collective DT strengths in an ensemble classifier to compensate for DT weaknesses (Breiman, L., 2001 - Pal M, 2005 - Cutler D, et al., 2007 - Belgiu M and L. Drăguţ, 2016 - He Y. and T.Warner, 2017). Averaging all tree votes determines each unknown's class. There is no longer concern that planting one tree may not be optimal. Thus, planting a forest should maximize global benefits (Maxwell A. et al., 2018). Bootstrapping builds each tree in the "forest." for resampling. At node splitting, a random subset of characteristics is selected for split variable selection. The majority vote and average determine categorization predictions for regressions (Breiman, 2001 - Yali, A and Donald G, 1997 - Tin K., 1995 - Tin Kam Ho, 1998). RF model tuning uses mtry and ntree. Splitting involves a random number of features (mtry) and several trees (ntree) to determine the output. Trade-offs exist for mtry. Large values make trees comparable but more accurate (Breiman, 2001). Use "Out of Bag" (OOB) samples—unused components—to validate your system. The mean of the trees' OOB-based projections is the result. RFs have a lot of flexibility and a high rate of being right. When the number of trees is considered, it also doesn't fit the data too well. On the other hand, as in DTs, a graphical representation is not possible (Resende P and Drummond A, 2018).

G. Naïve Bayes

The Naive Bayes method is a simple probability-band classifier sometimes known as the Bayes of Idiots, the Bayes of Freedom, or just plain old Bayes. To the extent that the class variable is specified, it is presumed that the presence or absence of a particular class feature is unrelated to the presence or absence of any other class feature (Oladipo I et al., 2022). The Naive Bayes is a simple method of estimating parameters because it does not utilize recursive algorithms. Therefore, a naive Bayes classifier may be of use when working with massive datasets. It also doesn't take a lot of data for training purposes to establish where the boundaries are. The assumption of independence among the variables allows for the estimation of variances rather than the entire matrix of covariance (Oladipo I et al., 2022).

1.2 Unsupervised Learning

Unsupervised learning involves datasets that do not contain any output label information. The objective of these models is to deduce the connection between data and/or to reveal latent variables (Singh G., 2018). High computational cost and multi-collinearity are two issues that can be avoided by limiting the number of features (Dormann F., 2013). During unsupervised learning, the machine makes an educated guess at the real-valued outcome by drawing on its own knowledge and previous encounters. Methods that rely on unsupervised learning include K-Means Clustering, Principal Component Analysis (PCA), and Latent Dirichlet Allocation (LDA) (Singh G., 2018).

A. K-Means Clustering

K-means clustering is a way to learn without being watched. It makes groups or clusters on its own. Data with similar traits are put together in the same cluster. The k-means clustering method has various uses. First, it's easier on a computer than hierarchical clustering with many variables. Second, it yields tighter clusters than global hierarchical clustering and small k. Finally, how simple the clustering results are to use and understand. The algorithm is computationally efficient because its complexity is O (K*n*d) (Ray S., 2019). However, K is uncertain and hard to determine. Global clusters impair performance because different initial partitions produce distinct final clusters. Performance suffers from input data cluster size and number volatility. Correlations between characteristics break the spherical assumption and give related features more weight, making it difficult to distribute features spherically inside each cluster. Outliers can hamper K-Means clustering. Any value of K has a different solution depending on beginning conditions and local ideals. This entails repeating the operation 20–100 times for any K number and selecting the solutions with the lowest J (Ray S., 2019).

B. Principal Component Analysis

PCA is an unsupervised machine learning technique that helps to simplify data. This means the calculations are faster and more reliable (Dey A., 2016). Using PCA, a set of variables can be transformed into a new, orthogonal set of variables known as principal components (PC). Data in two dimensions is reduced to a single dimension. Depending on the scale used, the outcomes of the PCA technique can vary. Therefore, scaling the data set is an essential step in the PCA algorithm (Dhall D and Juneja M., 2020).

C. Latent Dirichlet Allocation

Statistical data mining method LDA can characterize object classes in N-dimensional feature space using a sequence of values. This sequence represents the linear discriminant (McLachlan G., 2005). LDA and PCA are similar in finding the "most relevant" data differences and choosing routes that capitalize on them (I. T. Jolliffe et al., 1986). Labelled class variables distinguish LDA from PCA. How well the class means are divided along that axis compared to the overall class disparities determines its direction. This maximizes the interclass-intraclass dispersion ratio. It explores representations of data with fewer dimensions that cluster examples of the same class and separate instances of other classes (Gow J et al., 2012). Eigenvalues rank the k linear discriminants of the eigenvectors. You can use discriminants to classify new objects or reduce dimensionality (Gow J et al., 2012).

1.3 Reinforced Learning

RL is a goal-oriented technique, in contrast to supervised and unsupervised learning. Humans acquire knowledge through observation of their changing circumstances and responses to those alterations. To accomplish anything useful, RL must rely heavily on a learning agent (controller). Specifically, the agent acts (transmits control signals) to influence the environment's current state and receives unique values representing rewards and punishments. The goal of this agent is to maximize its reward as time progresses. A job provides a detailed description of the environment and elucidates the reward's origin (Coronato A et al., 2012). Instances of RL-based methods include the Q-Learning Algorithm and the Monte-Carlo Tree Search (MCTS).

A. Q-Learning

Using Q-learning (Watkin C & Dayan P.,1992), agents can quickly and easily learn optimal behavior in supervised Markovian environments. It's a low-resource, step-by-step approach to dynamic programming. The system is effective because it steadily improves its evaluations of the quality of certain acts in particular situations. It is possible to classify this technique as a form of Dynamic Programming (DP). Agents can gain insight into the optimal course of action to take in Markovian domains by observing the consequences of their actions without first having to construct detailed maps of those domains (Watkins C., 1989).

Q-learning applications in information theory are being studied. Q-learning and information theory are used in anomaly detection, pattern recognition, natural language processing (NLP), and picture classification (A. Achille and S. Soatto., 2018 - G. Williams et al., 2017 - J.T. Wilkes and C.R. Gallistel., 2017 - Jang B. et al., 2019). As a framework, RL may be used in a voice interaction system to offer a satisfying answer based on the user's speech (Y. An et al., 2017), while DL can predict local rainfall at high resolution (X. Shi et al., 2017). Ant Q-learning emerges in multi-agent settings where agents must agree to calculate a reward for a specific action. Ant Q-learning can get stuck at a local minimum if agents just take the fastest route (Chia-Feng Juang and Chun-Ming Lu., 2009).

B. Monte Carlo Tree Search

MCTS works well for sequential choices. The plan relies on intelligent tree search, which balances exploration and use. Random sampling in simulations helps MCTS track activity data and enhance decision-making with each iteration (Świechowski M., 2021). MCTS makes decisions using tree-based representations of large combinatorial spaces. The nodes of these trees represent problem states or configurations, and their connections show their transitions (Świechowski M., 2021). MCTS is officially used when a Markov Decision Process model can be created. MCTS can be used with POMDP by altering it (Lizotte DJ., 2016). Google DeepMind's AlphaGo (Silver D et al., 2016) is believed to have used MCTS and deep RL.

The MCTS technique is easy to grasp (Browne C et al., 2012). An uneven, systematic tree is built. Each iteration uses a tree policy to find the "root," or most important, node. A balance between discovery and exploitation is key to tree policy. The search tree is updated after a simulation from the node. It needs to update ancestor statistics and add a child node for the specified node's action. Simulation moves follow an unclear default rule. Randomness is the simplest strategy. MCTS minimizes the requirement to evaluate intermediate state values, reducing domain expertise (Browne C et al., 2012).

2. Big Data

The term "Big Data" was introduced in mid-2011 to describe huge, diversified data that is challenging to manage and handle using traditional tools and methods (C. K. Emani et al., 2015). Another definition of "Big Data" is the technology that lets scholars evaluate modern practices' massive data sets (R. L. Villars et al., 2011 - I. A. T. Hashem et al., 2015). BD includes volume, velocity, variety, veracity, and value. Data collected can reach tens or hundreds of petabytes. Velocity refers to the rapid delivery and accumulation of data and the increased need to respond quickly. This definition of "variety" includes semi-structured and unstructured data formats like audio, video, and text. Data like this often needs processing. Data veracity is correctness and reliability. Due to growing data volumes and sources, Big Data can be chaotic in the external world. The ultimate component is value (W. Jiang and J. Luo., 2022). Value is important since it quantifies data's utility (M. Arslan et al., 2017).

Processing data in real-time is a crucial concern in the field of Big Data. The fundamental aspect of BD streaming analytics is in the imperative to examine and react to real-time streaming data, such as traffic statistics, through uninterrupted queries to facilitate

instantaneous analysis during the data flow. The BD stream analysis process commences by ingesting data in the form of an endless tuple. It then proceeds to analyze the data and ultimately generates valuable outputs, typically in the form of an output stream (T. Kolajo et al., 2019).

2.1 Big Traffic Data Sources

This section explores the several data sources that can be employed for traffic prediction. Multiple methodologies exist for categorizing large traffic data. In the past, traffic data was classified into two types: unsupervised and supervised. Supervised data, such as loop detectors and Global Positioning System (GPS) traces, directly provide traffic information. On the other hand, unsupervised data, such as call detail records and cell phone location data indirectly provide traffic information that can be inferred. This evaluation will further classify large traffic data depending on its origins (Sayed S. A. et al., 2023).

Travel Surveys are extensive questionnaires used by municipalities or researchers to collect mobility data. Travel surveys provide comprehensive travel data, reducing the need for traffic forecasts. Travel surveys, albeit useful for real-time traffic forecast, have drawbacks. These include a small sample size, time and location limits, self-reporting inaccuracies, and high data collection expenses (Sayed S. A. et al., 2023). Public travel survey datasets include the Chicago Metropolitan Agency's "My Daily Travel Survey" (T. C. M. A.) and NREL's 2010–2012 "California Household Travel Survey (CHTS)" (T. S. D. Center).

Traffic Sensors data can be collected via radar, acoustics, infrared, and inductive loop detectors. Loop sensors are the most common road sensor technology. They offer many extensively used statistics for traffic prediction. Traffic sensor data can automatically and continually collect massive volumes of data in minutes or seconds. This data has poor spatial coverage and missing data due to insufficient sensors. Additionally, traffic sensor data on traffic flow speed may not be suitable for calculating road segment average journey time. Public traffic survey datasets include the Caltrans Performance Measurement System (PeMS) (Caltrans pems) and the Madrid City Council Open Data Portal's "Traffic Flow Madrid" (M. C. Council).

Electronic Toll Collection (ETC) that are widely utilized on toll highways and toll bridges, as well as in several other applications. The data collected from ETC systems is a thorough and efficient instrument for calculating motorway traffic. An important advantage of ETC systems is the extensive and cost-effective data collection they provide. However, the drawbacks are evident. Toll data can only be gathered in regions where ETC systems are operational (Sayed S. A. et al., 2023). The Amap data for Knowledge Discovery and Data Mining (KDD) CUP 2017 (Tianchi. Kdd cup 2017) is the sole publicly available toll ticket data known to the authors, which is utilized for the prediction of traffic flow.

2.2 Big Data Processing Frameworks

Traditional BD tools in the transportation sector are limited, thus the latest BD technologies have been used to analyze massive amounts of traffic data. Commercial tools have been used to improve spatiotemporal big data support. This section summarizes pre-existing BD processing systems used in traffic prediction jobs or that might be used for these goals. Big Data processing frameworks are either non-spatial or spatial. Non-spatial large data processing frameworks will be divided into Batch, Stream, and Hybrid categories. Figure 2 shows BD processing frameworks for each category (Sayed S. A. et al., 2023).

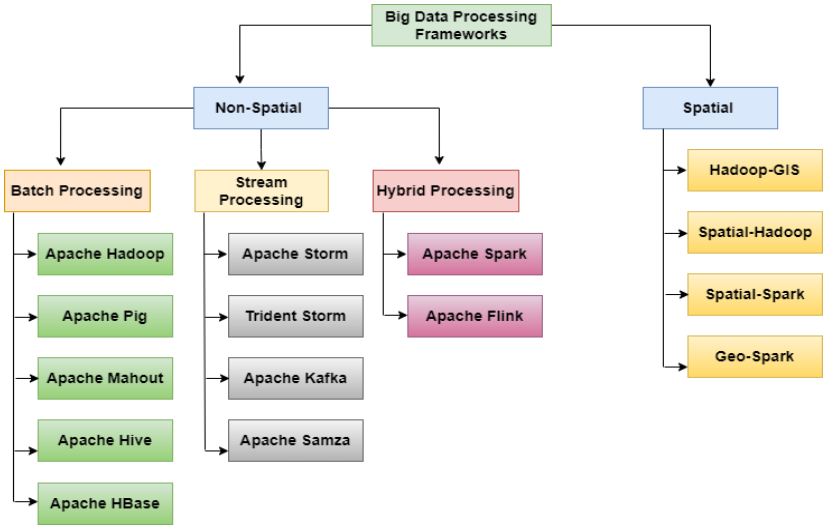

Fig. 2. *BD Processing Frameworks.*

III. SURVEY METHODOLOGY

The articles searched in this review were found in peer-reviewed publications from reputable publishers including Elsevier, Springer, IOP publishing, and IEEE. “Short-term traffic prediction” OR “traffic flow prediction” AND “machine learning” AND “traffic forecasting” OR “traffic speed forecasting” OR “Intelligent Transportation System” AND “Big Data Processing Frameworks”

AND “Big data in traffic flow prediction” are all search phrases that have been used to locate these articles. Prediction, performance evaluation, and the appropriate use of the appropriate techniques are all extensively covered in this paper. The survey's article searched clearly pertains to the topic of using ML methods to forecast traffic patterns. Both primary research articles and reviews were combed through to compile the data shown here.

1. ML Techniques for Traffic Flow Prediction

Predicting future traffic patterns is a crucial part of ITS work (Lippi M. et al., 2013). Many researchers in recent years have examined road traffic data using artificial intelligence, data mining, and statistical methods to forecast future traffic patterns (Aqib M. et al., 2019). As previous studies have shown, no single method is enough to evaluate such massive datasets. Therefore, appropriate technology must be utilized in accordance with the data's structure and volume to extract the most relevant insights (Janković S. et al., 2021).

Regression Model, a machine learning method, predicted traffic patterns in (Deekshetha H. R. et al., 2022). Matplotlib, Pandas, OS, Numpy, Sklearn, Tensorflow, and Keras support the model's regression model. "Traffic prediction" means to anticipate next year's traffic from this year's data. Precision and R2 are shown. Traffic statistics were calculated one hour later. The Kaggle dataset was used for this investigation. Two data sets exist. Some 2015 data includes date, time, number of cars, and number of intersections. The other is 2017 traffic statistics, which precisely match 2016. This study should use deep learning, big data, and other traffic flow prediction methods and examine more forecast accuracy factors.

Suneel Kuamr, 2022 investigated whether it would be possible to solve the problem of traffic regulation by employing an ML technique. The authors simulated traffic signals by employing the Q-learning RL method with an environment that they invented themselves, which they referred to as the Simulation of Urban Mobility (SUMO). The authors’ first set of experiments involved training a model with 300 neurons to determine all the potential parameters, including Cumulative Delay (CD), Average Queue Length (AQL), and Cumulative Negative Reward (CNR). In SUMO, the automobiles that are now in motion can have their delay times monitored, the delay times of individual vehicles can be controlled, and the delay times can be changed.

The main goal of (Navarro-Espinoza A. et al., 2022) was to outline an adaptive traffic control technique. To verify experiment reproducibility, Gradient Boosting Regressor, Linear Regression, MLP Regressor, RF Regressor, and Stochastic Gradient Descendent Regressor were utilized with all other settings set to default and the random state set to zero. In this paper, the authors used Huawei Munich Research Center Road Traffic Prediction Dataset from several traffic sensors. Performance criteria used to evaluate each ML were MAE, MAPE, RMSE, R2, and Explained Variance (EV) Score. When using MLP and Gradient Boosting, R2 and EV were above 0.93, MAE was 10.8, MAPE was 21%, and RMSE was 15.4. However, RF's R2 and EV scores were significantly below 0.93, its MAE 10.88, MAPE 21%, and RMSE 15.5. The RMSE was 15.85, MAE 11.2, and MAPE 24%. EV and linear regression R2 were 0.926. Conclusion: The stochastic gradient has an R2 value of 0.9, EV value of 0.9, MAE value of 12.8, MAPE value of 29%, and RMSE value of 18.

(Upadhyaya S. and Mehrotra D., 2022) integrated four ML models—NB, SVM, DT, and RF—to predict traffic flow lane changes, which is crucial to ITS success. Federal Highway Administration (FHWA) used "Next-Generation Simulation (NGSIM)" to collect a high-fidelity vehicle flow dataset in 2005. Trajectories provided accurate lane designations and vehicle positions. This study uses these data. Data was used to assess model lane-change forecast accuracy. SVM predicted car lane changes better than the other three ML models.

Using the fuzzy logic framework and a time series of urban traffic volumes, the authors of (Qu Z. and Li J., 2022) proposed a type-2 fuzzy logic-based approach to prediction. Interval type-2 fuzzy system predictions were improved with the help of the Back Propagation (BP) technique, which involved updating antecedent coefficients and fuzzy rule outputs. Data on transportation networks were used to assess the method described in this study and compare it to other fuzzy methods. Predictions made with the BP technique and SVM trained using the type-2 fuzzy logic system are more accurate.

In (Steffen T. and Lichtenberg G., 2022), the authors examined historical data-based road traffic prediction methods. This approach uses traffic data convolution polygonal (CP) tensor decomposition. Extraction of daily and weekly features and traffic spatial distribution considerably decreases the quantity of data needed to explain a traffic signal. Deconstructing and moving the key components forward yields traffic data. Starting October 1, 2019, 15-minute data gathering sessions was place on Northern England's M62 until October 28, 2019. Data is shown as vehicles per hour. The four-parameter forecast method outperforms cutting-edge rolling-average prediction algorithms.

An ML-based intelligent traffic monitoring system (ML-ITMS) was developed in (Wang J. et al., 2022) to predict roadside traffic congestion and improve ITS. ML was used to model the current situation, and SVM parameters were adjusted to improve the short-term traffic forecast. The suggested ML-ITMS generated high-capacity, wide-area networks with a single SVM-RF query. The proposed ML-ITMS improved traffic flow and nonparametric process predictions with mathematical models. The proposed ML-ITMS was empirically tested. ML-ITMS applications included automatic parking, LIDAR-based traffic predictions, citywide security monitoring, and healthcare delivery. The results reveal that the proposed ML-ITMS outperforms state-of-the-art approaches in traffic tracking (up to 98.6% accuracy) and traffic flow prediction.

A geophysical-informed search approach for an extreme learning machine (ELM) was proposed in (Cui Z. et al. 2022). Short-term traffic forecasts are being considered as a potential answer. ELM avoids BP's tedious process by zeroing in on the best possible solution. Taking everything into account, the suggested search approach aims to find the best possible settings for ELMs. Many recent models have been used to assess how well the proposed search approach predicts on four standard data sets. The real traffic on the A1, A2, A4, and A8 freeways that make up the Amsterdam Ring Road was used to create the four reference datasets. On these datasets, the GSA-ELM model obtains MAPEs of 11.69%, 10.25%, 11.72%, and 12.05%, while the RMSEs fall in the 288.09%-163.24% range.

Supervised ML was investigated as a Big Data analytics technique for traffic volume forecasting in (Janković S. et al., 2021). Two case studies were used to determine this. Both sets of studies employed training and testing data gathered from prediction models fed traffic counts collected by selected automatic traffic counters deployed on roadways in the Republic of Serbia between 2011 and 2018. In the first study, models based on DT got the best results. In the second study, models based on Lazy IBk, RF, Random Committee, and Random Tree algorithms got some of the best results. In the first case study, the model based on the M5P algorithm demonstrated the highest performance, whereas the Lazy IBk method produced the best results in the second study.

To organize traffic, the authors of (Li J. et al., 2021) developed an ML method for estimating journey times. They tested the Gaussian Process Regressor (GPR) for this. After determining the average travel time for a route, "clustering" shows that different "kinds of days" have varied daily travel time profiles. Then, numerous regression parameters estimated traffic volume from journey times. The authors examined two cases in this study. The regression component for each day's profile was trained using "multi-model" variance. This "Single Model" variation trained one Regressor without considering the day profile. The unique route network traffic flow prediction and reestablishment strategy uses ML and automobile data. This study illuminates two main difficulties. First, training with non-dispersed algorithms might complicate and lengthen assessment sequences. Second, Every ML-based approach has a second issue regarding data quality relevance.

In (Knapińska A. et al., 2021), the authors examined online multiple-time series prediction for traffic of different frame sizes. The model predicted traffic using previous data. The authors began by explaining how they collected and analyzed real-world network traffic data to find periodicity and relationships across traffic types. They also used linear regression, KNN, and RF to predict network data under different models and input features. Linear regression has 50% lower RMSPE than KNN and 15% lower than RF. The authors also examined how different models and input features might affect the outcome, finding the best speed-accuracy balance.

In (Zeroual A. et al., 2021), traffic flow predictions were made using a kernel-based learning model called support vector regression (SVR). To evaluate the SVR model's ability to predict traffic densities, the authors looked into a variety of kernels. Data collected from California motorways was used to assess the SVR's effectiveness. The authors also provided a comparison between linear regression, linear SVR, quadratic SVR, and gaussian SVR. Results from this study supported the use of the Gaussian SVR for predicting traffic flows. These findings demonstrate that the Gaussian kernel SVR model is superior to the alternatives.

In (Ramchandra N. R. and Rajabhushanam C., 2021), many well-established ML methods, such as Deep Autoencoder (DAN), Deep Belief Network (DBN), and RF, were applied to an online data set to forecast traffic flows. It was possible to foretell the zone's traffic volume by using the four most important characteristics of time of day, season, temperature, and location. Accuracy, precision, RMSE, and MSE values were used to assess the proposed system's efficacy. The RF method is the most effective of the three with an accuracy of 92.6%, MSE of 346.35, and RMSE of 18.61.

Global and cluster-based methods for learning vehicle speed prediction in large, dynamic sensor networks were given in (Magalhaes R. P. et al., 2021). The authors evaluate and contrast prediction models trained with the three techniques to answer three experimental questions. The authors employed a large real-world sensor dataset and cutting-edge ML methods to train the model. We tested multivariate linear regression (MLR), radial basis functions (RF), and gradient boosting regression trees (GBRT). Over the dataset's 12 months, 130 sensors were added to the system. Developing global models for dynamic sensor network difficulties has shown success. Researchers received the real-world dataset necessary to validate their findings. Fortaleza, Brazil, had 272 road traffic sensors that collected 1.3 billion data points in 2014. This was the study's dataset. Historical Average (HA), a baseline method for estimating average speed in a time interval by averaging all training example speeds, was also studied.

Trend-based Online Network Traffic Analysis (TONTA), a new pattern recognition method for ad-hoc IoT networks, was suggested in (Shahraki A. et al., 2021) to monitor network performance. A statistically light method called Trend Change Detection (TCD) was used online in the proposed method. To analyze the traffic on the IoT network, TONTA looks for major trends and notices sudden or gradual changes in time-series datasets. As an offline benchmark TCD technique, Relative unconstrained Least-Squares Importance Fitting (RuLSIF) was used to compare how well TONTA worked with how well it worked. The results showed that TONTA finds about 60% fewer false positive alarms than RuLSIF.

A hybrid model using ELM and ensemble-based approaches was constructed to forecast future hourly traffic on a road stretch in Tangiers, northern Morocco (Jiber M. et al., 2020). SLFN and other fast ML algorithms were employed to develop their strategy. Moroccan Center for Road Studies and Research annualized traffic estimates from 2013 to 2017 were used to complete the work. Consider all factors that affect road traffic flow to gain a complete picture. Conditions like weather and highway layout are examples. The suggested model was compared to MLP, SVR, and ARIMA standard algorithms. Simulations revealed good precision and stability.

Four ML models—RF, SVR, MLP, and MLR—were tested for predictive power using road network data from Thessaloniki, Greece (Bratsas C. et al., 2020). A vast collection of stationary and mobile traffic data covered the metropolis in space and time. The authors summarized 15-minute periods of numbers. They standardized the number of cars entering each route, its minimum and highest speeds, standard deviations, means, skewnesses, and kurtosis, and used these statistics to train ML algorithms that forecast average speeds. The optimal model parameters were found using 10-fold cross-validation. Splitting the training dataset into 10 similar datasets allowed 10-fold cross-validation. The requested mean speed estimates were made by training the RF, SVR, and MLP on 9 datasets. The SVR model works best in stable, low-change contexts, whereas the MLP model is more flexible and has near-zero errors in high-change environments.

The authors of (Meena G. et al., 2020) devised an accurate and timely traffic flow prediction algorithm. Combining ML, genetic, and soft computing methods simplified transportation system data analysis. Traffic sign recognition, a key stage in autonomous vehicle training, used image processing. The paper's goals were met utilizing Android Studio, Java, Garmin, PHP, XML, Python, and sklearn. The authors constructed and analyzed many ML methods to improve productivity and precision. The DT algorithm predicted target variable values (88% accuracy, 108.4 sec). They used SVM (Accuracy: 88%, Time: 94.1 sec) to locate the most extreme data for a precise forecast. Finally, the RF method (Accuracy: 91%, Time: 110.1 sec) predicted traffic congestion.

In (Zheng L. et al., 2020), the authors proposed a gradient-boosted regression tree (DSTO-GBRT) based on dynamic spatial-temporal characteristics for short-term traffic flow prediction using Electronic Registration Identification (ERI) big data, a new vehicle identification technology. First, the DSTO-GBRT architecture was created. The spatial-temporal relationships between the current forecast point and upstream correlative locations were examined using Pearson's correlation coefficient. PCA minimized linear correlations among features to refine the initial training data and generate high-quality data. The authors tested the model using Chongqing ERI data from March 1 to March 31, 2016. Test data was used on March 25, 2016, and training data was used elsewhere. Compared to ST-GBRT, ARIMA, DSTO-BPNN, and DSTO-SVM, DSTO-GBRT provided a rapid and adaptable prediction, especially during rush hour when traffic conditions change quickly. The proposed DSTO-GBRT method surpassed DSO-GBRT and DTO-GBRT in accuracy.

In (Kamble S. J. and Kounte M. R., 2020), multiple characteristics, including hard delay limitations and the speed available across a GPS vehicle's route were used to identify traffic congestion using the ML technique. To estimate traffic speeds, they have implemented the Gaussian Regressor process in ML, which makes use of three datasets: the training set, the prediction set, and the road sector data frame for historical route networks data. The authors have used the data set to determine three distinct periods during which to monitor traffic congestion and have analyzed the average speed of cars on the road sector during each of these periods.

A revolutionary architecture predicted city traffic flow, according to (Inzunza M. C. et al., 2020). ML, Computer Vision (CV), DL, and neural networks were considered for the answer. The Machine Algorithm System (MASY) analyzes traffic patterns, the Neuronal Artificial System (NASY) classifies traffic, the Web user application (WeUsAP) displays results and processes user data entry, and the Car Counting Wizard (CCW) captures video using computer vision to generate a statistical analysis of vehicle counts. The approximation was accurate, the model improved over time, and the forecast model matched CCW's automotive output.

The authors of (Weerasekera R. et al., 2020) tested ANN, RF, and SVR algorithms for simulating traffic flow at various data resolutions and responding to unexpected traffic accidents. To focus on spatiotemporal attributes that most affect model accuracy, several feature selection methods have been examined. These tests showed that multivariate spatiotemporal ML models with aggregated data are not always effective. Models learned with high-resolution 30-second input data outperformed baseline ARIMA models by 8%. An added benefit was that Recursive Feature Elimination-based feature selection beat linear correlation-based models. The authors used Auckland roadway traffic and occupancy information to derive conclusions.

The authors of (Zahid M. et al., 2020) used state-of-the-art models and hyperparameter optimization to accurately simulate short-term future traffic condition prediction using ML classifiers such local deep SVM (LD-SVM), decision jungles (DJ), MLP, and CN2 rule induction. Level of service (LOS) horizons and simple if-then rules are also used to assess traffic conditions over time. They found that random hyperparameter tweaking produced the best results. Decision jungle and LD-SVM improved prediction accuracy by 0.982 and 0.975, respectively, for a roughly 95% improvement. The experiments indicated that DJs were more trustworthy and productive than competitors. VISSIM was used to simulate a basic section of Beijing's Second Ring Road expressway and calculate short-term traffic status as a function of level-of-service (LOS). A roadway length's real density flow was calculated using 15, 10, and 5-minute prediction times.

A hybrid predicting model that combines decomposition and prediction was introduced in (Wang Z. et al., 2020) to improve highway traffic flow estimation. The California Department of Transportation's Performance Measurement System detector VDS-1209092 on Irvine's I405-N motorway provided the data. They trained and adapted prediction models using 2016 data points from May 1 to May 7, 2019, including speed, occupancy rate, and traffic flow. Every data set was split in two. This model was trained using data from May 1–5. The model's settings were adjusted on May 6 and 7, 2019. After training the model, the day's traffic volume was predicted and compared to May 8, 2019. The full ensemble empirical mode decomposition with adaptive noise (CEEMDAN) approach adaptively decomposed complicated, nonlinear highway traffic flow data. Improved weighted permutation entropy was used to reconstruct new elements. The least-squares SVM (LSSVM) prediction model they created for each reconstruction segment

included the subsequence predictions, making it more accurate. The authors' experiments showed that the model helps anticipate traffic flow and analyze trends, helping transportation officials make better decisions.

Intelligent Internet of Things (IIoT) networks are dynamic due to their ever-changing topology and large range of services, making traffic forecasting difficult. These observations encouraged the authors in (Nie L. et al., 2020) to propose a reinforcement learning mechanism. They predicted network traffic using a Markov decision process model and refined it with Monte-Carlo Q-learning. They also devised a residual-based dictionary learning technique to simplify Monte-Carlo Q-learning and made the process real-time. The authors tested the RL-based network traffic prediction algorithm on a 12-node testbed. The constructed wireless network supports video, telephony, and other services. The generic urban path loss model sets OSPF weights, and the open shortest path first (OSPF) algorithm creates the testbed's topology. Their proposed technique was tested using real network traffic.

Clustering and series models were combined in (Aldhyani T. H. et al., 2020) to improve network traffic prediction. They improved our time series using fuzzy c-means clustering granules. The suggested model was tested using real network data (4G Cell Traffic from Kaggle and Measurement and Analysis on the Wide Internet) from diverse network backbones. The proposed research uses clustering to handle ambiguity in entire network data, which is better than time series models. For even greater model accuracy, the authors recommend preprocessing using a weighted exponential smoothing model. The authors claimed AI could predict network user behavior. AI is preprocessing time series models to improve them. In the Adaptive Neuro-Fuzzy Inference System (ANFIS), non-crisp Fuzzy-C-Means (FCM) clustering and weight exponential technique produce better time series models. To predict network traffic, ANFIS time series model was constructed. Two snapshots of network traffic were used to evaluate the suggested time series models. ANFIS model empirical tests utilizing cellular traffic data showed a correlation indicator R of 96.78%. The suggested model outperforms competing time series models.

EL was used to update the distributional representation in (Xiao J. et al., 2019), a new stepwise regression method in an ideal drift environment based on Learn++ for SVR. The author first converted the regression problem into a binary classification problem to estimate future foot traffic. Second, the R2C technique improved classification-style ensemble learning loss functions. Next, represent hyper-resolution improvement as incremental regression function learning. Since motion volume is spatially dependent, the R2C architecture's motion volume prediction is flawed. This study used data from the Caltrans PeMS, which receives traffic data from over 15,000 detectors every 30 seconds. California's motorways had detectors everywhere.

A new ITS infrastructure approach was suggested in (Rajkumar S. C., 2019). This approach used a magnetic sensor to count and classify automobiles in a traffic pattern. The cluster then delivers data to a cloud server using MapReduce and local proximity services. An intelligent cloud server agent using the Markov Decision Process (MDP) reinforcement learning algorithm predicted the best route for registered users. This study employed a real-time dataset from 1 Nov to 15 Nov 2018. Applied successfully, this approach has a 98.36% success rate.

In (Tu M. et al., 2019), the authors introduced a new traffic prediction method using least squares SVM and K-means clustering algorithms for data-driven traffic management and decision-making. Historical traffic statistics came from Nanchang's Donghu District. The LSSVM algorithm has a prediction error rate of 25.33 %, while K-means clustering had 20%. Due to traffic flow data anomalies, transportation system complexity, and data incompleteness, there are still many defects. (1) Traffic flow data volume and connection. (2) Flow, vehicle speed, and occupancy aspects of traffic flow data are interdependent. (3) The model ignores abrupt events like traffic accidents and extreme weather. The specific qualities of highways, trails, and national roads, which should be considered, were not researched.

Traffic-noise discomfort was predicted using ML models (Bravo-Moncayo L. et al., 2019) that incorporated noise perception, exposure, and demographics. ANN, SVM, and MLR were used to create traffic noise nuisance models and compare their error rates. City building data was obtained from the Municipality database registration for this investigation. From 2010 to 2016, the Municipal Mobility Secretariat conducted one-week automatic car counts on 523 streets on various roadways. The case study area's traffic noise exposure was mapped and estimated. Since an individual's perception of noise and expected exposure to noise affect traffic-noise discomfort, traditional statistical models cannot provide trustworthy forecasts. Implementing an ML technique enhanced accuracy and R2 values. The ANN model predicted traffic-noise annoyance better than the MRL and SVM by 42% and 35%, respectively, in training subsets. Test dataset subsets had 24% and 19% less error. ANN had 3.8- and 2.3-fold R2 gains in training datasets and 1.7-fold increases in testing datasets compared to MLR and SVM models. In this study, MatLab and R were used to tackle classification and regression issues using MLA.

In (Li Y. and Jiang W., 2019), big data short-term traffic flow prediction accuracy and timeliness were examined. The Caltrans PeMS contributed this study's data. Traffic data is collected in real time from over 39,000 detectors. This report uses data from Pasadena Road Monitoring Point 771668 from January 1, 2010 to January 1, 2019. This study's data samples use a 5-minute sale detection time. Cluster analysis and Regression Forest were used to develop a Spark prediction model. It used K-means cluster analysis to assess traffic volume and climatic conditions over time and space. The parallel Regression Forest method was used to predict and train massive data sets on a spark distributed computing cluster. Testing showed that the integrated forecast model for comparable data sets runs faster on a Spark cluster. The benefit grows with data volumes. When employed alone, it outperforms SVM and regression forest in forecast accuracy.

In (Fang C. et al., 2019), the authors found that identifying road traffic statuses significantly affected short-term traffic flow forecast accuracy. This study uses data from six RTMS on Beijing's second ring road during December 1–5, 2003. The experiment's RTMS detector is located on Beijing's four-lane Second Ring Road. A workday is the timeframe. The authors anticipate traffic flows using ARMA and Kalman filters in their study. The suggested method splits measured traffic data for each road traffic condition in half and compares the halves using expected outcome indices. Finally, many traffic forecasts and flow advice were given. This study found that state partitioning drastically reduces ARMA and Kalman filter short-term traffic flow forecast accuracy. A plus is that an ARMA model can be adjusted to increase forecast accuracy when road traffic tidal characteristics are clear and lane conditions are the same. Predictions and results may vary depending on state partitioning approach.

In (H. Mehdi et al., 2019), cloud traffic was predicted using a method called fuzzy autoregressive integrated moving average (FARIMA), which combines ARIMA with fuzzy regression. In this study, the authors used a portion of the Wideadjp dataset for simulation. From November 1 to December 30, 2015, Wideadjp received 228 GB requests. Better prediction accuracy can be attained with the use of fuzzy ARIMA (SOFA) models. RMSE and coefficient of determination comparisons show that SOFA is the more accurate model for traffic prediction, with values of approximately 5.4 and 0.009, respectively.

Backpropagation bidirectional ELM (BP-BELM) models are introduced in (Zou W. and Xia Y., 2019) as a unique prediction model. To fine-tune back propagation settings, prior experience is not required. In this study, the authors used eight UCI datasets (AutoMPG, Automobile, DrivFace, Fertility, NoisyOffice, Servo, UJIIndoorLoc, and wiki4HE). In the end, the simulations, and comparisons show that BP-BELM is more accurate at predicting traffic flow than other methods, such as back propagation neural networks, radial basis functions, SVM, and others that have been developed incrementally.

The authors of (Yang H. et al., 2019) employed the powerful and systematic Taguchi approach to optimize the proposed exponential smoothing and ELM forecasting model and find the optimal configuration. The new model was tested on data collected from UK motorways and freeways, and the results were compared to those of three other forecasting models. With an accuracy of about 91% and 88% during peak and off-peak traffic times on the freeway and highway, respectively, the results showed that the Taguchi technique is a great way to design a forecasting model and that the proposed model with its optimized configuration does a better job of predicting traffic flow. To summarize all previous related works, Table 1 compares them in terms of methodology, data set, adopted techniques, and results.

TABLE II. SUMMARY OF RELATED WORKS.

No.	Methodology	Dataset	Techniques	Results
(Deekshetha H. R. et al., 2022)	ML algorithm to forecast next year's traffic using data from the previous year.	Two Kaggle datasets.	Regression Model	N/A.
(Suneel Kuamr, 2022)	Traffic signals simulation with an environment called SUMO.	N/A.	Q-learning RL method.	SUMO monitors, controls, and changes delay times
(Navarro-Espinoza A. et al., 2022)	A strategy for adaptive traffic control.	Huawei Munich Research Center Dataset.	Linear Regression, MLP Regressor, Gradient Boosting Regressor, RF Regressor, and Stochastic Gradient Descendent Regressor.	R^2: of 0.9, EV: 0.9, MAE: 12.8, MAPE: 29%, and RMSE: 18.
(Upadhyaya S. and Mehrotra D., 2022)	Predict lane changes in traffic flows.	High-fidelity vehicle flow dataset in 2005.	SVM, NB, RF, and DT.	SVM has the highest accuracy.
(Qu Z. and Li J., 2022)	Time series analysis of urban traffic volumes prediction	Data on transportation networks.	Interval type-2 fuzzy logic, BP, and SVM.	Predictions with type-2 fuzzy logic system are more accurate.
(Steffen T. and Lichtenberg G., 2022)	Predicting traffic along a certain road based on past data.	Northern England's M62 dataset.	CP tensor decomposition of traffic data.	The proposed forecast method significantly beats the competition.
(Wang J. et al., 2022)	ML-ITMS to predict congestion at roadside sensors.	N/A	SVM and RF.	Accuracy 98.6%
(Cui Z.et al., 2022)	Short-term traffic predictions based on ELM.	Real traffic data on the Amsterdam Ring Road	ELM.	**MAPEs:** of 11.69%, 10.25%, 11.72%, and 12.05%, **RMSEs**: 288.09% to 163.24%
(Janković S. et al., 2021)	Supervised ML traffic volume prediction.	Counts of vehicles on the roads in the Republic of Serbia.	DT, Lazy IBk, RF, Random Committee, and Random Tree algorithms.	- M5P and Lazy IBk have the best results
(Li J. et al., 2021)	An ML technique for predicting journey times.	N/A.	GPR.	N/A.
(Knapińska A.et al., 2021)	An online multiple-time series traffic prediction.	N/A.	Linear Regression, KNN, and RF.	Linear Regression has highest accuracy.
(Zeroual A. et al., 2021)	SVR model to forecast traffic flow.	Caltrans PeMs.	Gaussian SVR.	Gaussian SVR has the best performance.
(Ramchandra N. R. and	Forecast traffic flows using ML methods.	An online dataset.	DAN, DBN, RF.	RF: 92.6% accuracy.

Rajabhushanam C., 2021)				
(Magalhaes R. P. et al., 2021)	Global and cluster-based learning for vehicle speed forecasting.	A sizable sensor dataset spans 12 months.	MLR, RF, GBRT.	HA was investigated
(Shahraki A. et al., 2021)	TONTA to monitor network performance using TCD.	Traffic data on the IoT network.	TCD, RuLSIF.	TONTA: 60% fewer false positive than RuLSIF.
(Jiber M. et al., 2020)	Hourly traffic forecasting.	Annualized traffic numbers from 2013 to 2017.	SLFN	The proposed model offers higher performance.
(Bratsas C. et al., 2020)	Evaluate the predictive skills of some ML models.	Data from the road network in Thessaloniki, Greece.	RF, SVR, MLP, and MLR.	SVR and MLP have the best performance.
(Meena G. et al., 2020)	ML, genetic, and soft computing approaches for predicting traffic flow information.	Traffic signs recognized using Image Processing techniques.	DT, SVM, RF	**RF**: Accuracy 91%, Time 110.1 sec.
(Zheng L. et al., 2020)	DSTO-GBRT to predict traffic flows	Real-world big ERI data from March 1 to March 31 of 2016	GBRT, Pearson's correlation coefficient, and PCA.	DSTO-GBRT has the best performance
(Kamble, S. J. and Kounte M. R., 2020)	ML technique to identify traffic congestion.	Road sector data frame for historical route networks data.	Gaussian Regressor.	The authors have analyzed the average speed of cars on the road sector.
(Inzunza M. C. H. et al., 2020)	A novel architecture was built to anticipate traffic flow.	N/A.	ML, CV, DL, and neural networks.	The model has heigh accuracy.
(Weerasekera R. et al., 2020)	ML techniques to model traffic flow.	Traffic records from a roadway in Auckland (New Zealand).	ANN, RF, and SVR.	The proposed models have heigh accuracy
(Zahid M. et al., 2020)	Cutting-edge models and hyperparameter optimization to predict traffic conditions.	A simple part of the Second Ring Road freeway in Beijing, China.	LD-SVM, DJ, MLP, and CN2 rule induction.	Decision jungle and LD-SVM have the best performance.
(Wang Z. et al., 2020)	Hybrid predicting model for traffic prediction.	Caltrans PeMS Dataset.	LSSVM.	The model aids in traffic flow forecasting and trend analysis.
(Nie L. et al., 2020)	RL-based network traffic prediction method.	A testbed with 12 nodes to test the proposed model	RL, Q-Learinig.	An analysis of actual network traffic was used to test the model.
(Aldhyani T. H. et al., 2020)	ANFIS time series model was created for network traffic prediction.	4G Cell Traffic data from Kaggle.	Fuzzy-C-Means clustering.	correlation indicator R: 96.78%.
(Xiao J.et al., 2019)	A new framework based on stepwise regression for traffic prediction.	Caltrans PeMS.	Learn++ for SVR, R2C method,	R2C architecture's low accuracy
(Rajkumar S. C., 2019)	A new ITS infrastructure to determine both the vehicles count and their types in traffic pattern.	A real-time dataset from 1 Nov to 15 Nov 2018.	RL.	**RL**: 98.36% accuracy.
(Tu M. et al., 2019)	A new method for traffic prediction.	Historical traffic data from Nanchang's Donghu District.	LSSVM and K-means clustering	**LSSVM**: error rate 25.33 %, **K-Means:** error rate 20 %.
(Bravo-Moncayo et al., 2019)	Traffic-noise discomfort was predicted using ML models.	Information about buildings in cities from the Municipality database register.	ANN, SVM, and MLR.	The ANN model has the highest accuracy.
(Li Y. et al., 2019)	traffic flow predictions using Spark cluster.	Caltrans PeMS.	K-means clustering.	the proposed model has a high accuracy
(Fang C. et al., 2019)	The precision of short-term traffic flow forecasts was investigated.	Information gathered by six RTMS on Beijing's second ring road.	ARMA, Kalman filter technique	prediction accuracy of the ARMA and Kalman filter models affected by state partitioning.
(H. Mehdi et al., 2019)	Cloud traffic was predicted using FARIMA.	A portion of the Wideadjp dataset for simulation.	ARIMA, fuzzy regression.	**RMSE and coefficient of determination**: 5.4 and 0.009, respectively.
(Zou W. and Xia Y., 2019)	ELM (BP-BELM) models were developed as a unique prediction model.	Eight UCI datasets	BP-BELM.	BP-BELM has a high accuracy.
(Yang H. et al., 2019)	ELM forecasting model	Data from UK motorways and freeways.	ELM.	ELM: 91% Accuracy.

IV. CHALLENGES AND OPPORTUNITIES

Traffic bottlenecks can be dangerous, especially in heavily populated places, thus planning and predicting traffic is crucial. Reliable and effective road traffic prediction methods are needed. This study addresses the absence of computer-friendly traffic flow forecasting methods and algorithms. High-quality training data is scarce. Network models were trained using incomplete data from matched city traffic flow statistics. If they are true, ML can't estimate traffic flow as well. ML's underuse of real-time spatio-temporal correlations causes the gap. The complicated linkages between road segments and congestion patterns explain this. Lack of processing power and centralized storage complicates traffic forecasts. This needs further study.

The current study is limited to the processes and algorithms in the examined literature. It's possible this investigation overlooked some techniques. The literature study covers powerful DL methods like CNN and LSTM, which should be studied further. This is now

achievable since traffic data from different cities may be utilized to train models on more universal data patterns. Thus, ML and DL algorithms will improve traffic projections in smaller cities. Communicating with the city's urban administration to deliver a mountain of relevant big data will be the researchers' biggest challenge. Legal constraints on sharing traffic data to municipal governments can provide challenges. Traffic-monitoring sensors fed into machine-learning algorithms for better decision-making may increase linked, high-risk IoT environments. Cybersecurity threats should be anticipated in ITS cities. This raises several research questions.

There have been many successful Big BD traffic prediction applications, however there are still certain challenges. The data density varies widely between modes of transportation, and data shortage, excessive missing data, distortion, and deficiency persist. Data quality, privacy, and policies have been understudied. This section discusses possible solutions. Crowdsourced data has poor quality, noise issues, and privacy concerns. To circumvent these issues, sparse BA has been used to predict traffic conditions using under-sampled data (C. N. Babu et al., 2019).

V. CONCLUSION

The objective of this paper was to provide a thorough analysis of the prominent ML approaches employed in traffic forecasting, while also examining the latest advancements in big data in traffic prediction. The focus was on the processes of traffic prediction and the suitability and effectiveness of ML and BD as viable options. Additionally, it provided an examination of the many sources of big data that can be employed in traffic forecasting activities. Furthermore, it addressed the challenges linked to the utilization of BD and ML in traffic prediction. As a result of this review, a total of 36 papers were selected and thoroughly investigated after a rigorous selection procedure.

REFERENCES

[1] A. Achille and S. Soatto, "Information dropout: Learning optimal representations through noisy computation," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018.

[2] Aldhyani, T. H., Alrasheedi, M., Alqarni, A. A., Alzahrani, M. Y., & Bamhdi, A. M. (2020). Intelligent hybrid model to enhance time series models for predicting network traffic. IEEE Access, 8, 130431-130451.

[3] An, S., Lee, B.-H., & Shin, D.-R. (2011). A Survey of Intelligent Transportation Systems. 2011 Third International Conference on Computational Intelligence, Communication Systems and Networks.

[4] Aqib, M.; Mehmood, R.; Alzahrani, A.; Katib, I.; Albeshri, A.; Altowaijri, S.M. 2019. Smarter Traffic Prediction Using Big Data, In-Memory Computing, Deep Learning and GPUs, Sensors 19: 2206.

[5] Belgiu, M., and L. Drăguţ. 2016. "Random Forest in Remote Sensing: A Review of Applications and Future Directions." ISPRS Journal of Photogrammetry and Remote Sensing 114: 24–31.

[6] Bengio, Y., Learning deep architectures for AI. 2009: Now Publishers Inc.

[7] Bratsas, C., Koupidis, K., Salanova, J. M., Giannakopoulos, K., Kaloudis, A., & Aifadopoulou, G. (2020). A comparison of machine learning methods for the prediction of traffic speed in urban places. Sustainability, 12(1), 142.

[8] Bravo-Moncayo, L., Lucio-Naranjo, J., Chávez, M., Pavón-García, I., & Garzón, C. (2019). A machine learning approach for traffic-noise annoyance assessment. Applied Acoustics, 156, 262-270.

[9] Breiman, L. 2001. "Random Forests." Machine Learning 54 (1): 5–32.

[10] Breiman. 2001. Random forests. Machine Learning 45, 1 (Oct. 2001), 5–32.

[11] Browne, C. B., Powley, E., Whitehouse, D., Lucas, S. M., Cowling, P. I., Rohlfshagen, P., ... & Colton, S. (2012). A survey of monte carlo tree search methods. IEEE Transactions on Computational Intelligence and AI in games, 4(1), 1-43.

[12] C. K. Emani, N. Cullot, and C. Nicolle, "Understandable big data: a survey," Computer science review, vol. 17, pp. 70–81, 2015.

[13] C. N. Babu, P. Sure, and C. M. Bhuma, "Sparse bayesian learning assisted approaches for road network traffic state estimation," IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 3, pp. 1733–1741, 2020.

[14] Caltrans. Caltrans performance measurement system (pems). [Online]. Available: (http://pems.dot.ca.gov/).

[15] Charbuty, B., & Abdulazeez, A. (2021). Classification based on decision tree algorithm for machine learning. Journal of Applied Science and Technology Trends, 2(01), 20-28.

[16] Chen, C., Li, K., Teo, S. G., Zou, X., Li, K., & Zeng, Z. (2020). Citywide traffic flow prediction based on multiple gated Spatio-temporal convolutional neural networks. ACM Transactions on Knowledge Discovery from Data (TKDD), 14(4), 1–23.

[17] Chen, K., Chen, F., Lai, B., Jin, Z., Liu, Y., Li, K., Wei, L., Wang, P., Tang, Y., Huang, J., Hua, X. (2020). Dynamic Spatio-temporal graph-based CNNs for traffic flow prediction. IEEE Access, 8, 185136–185145.

[18] Chia-Feng Juang and Chun-Ming Lu. 2009. Ant colony optimization incorporated with fuzzy Q-learning for reinforcement fuzzy control. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans 39, 3 (2009), 597–608.

[19] Chowdary, G. J. (2021). Machine Learning and Deep Learning Methods for Building Intelligent Systems in Medicine and Drug Discovery: A Comprehensive Survey. arXiv preprint arXiv:2107.14037.

[20] Coronato, A., Naeem, M., De Pietro, G., & Paragliola, G. (2020). Reinforcement learning for intelligent healthcare applications: A survey. Artificial Intelligence in Medicine, 109, 101964.

[21] Cui, Z., Huang, B., Dou, H., Tan, G., Zheng, S., & Zhou, T. (2022). GSA-ELM: A hybrid learning model for short-term traffic flow forecasting. IET Intelligent Transport Systems, 16(1), 41-52.

[22] Cutler, D. R., T. C. Edwards Jr., K. H. Beard, A. Cutler, K. T. Hess, J. Gibson, and J. J. Lawler. 2007. "Random Forests for Classification in Ecology." Ecology 88 (11): 2783–2792.

[23] Deekshetha, H. R., Shreyas Madhav, A. V., & Tyagi, A. K. (2022). Traffic Prediction Using Machine Learning. In Evolutionary Computing and Mobile Sustainable Networks (pp. 969-983). Springer, Singapore.

[24] Dey, A.: Machine learning algorithms: a review. Int. J. Comput. Sci. Inf. Technol. 7(3), 1174–1179 (2016).

[25] Dhall, D., Kaur, R., & Juneja, M. (2020). Machine learning: a review of the algorithms and its applications. Proceedings of ICRIC 2019, 47-63.

[26] Dormann, C. F., Elith, J., Bacher, S., Buchmann, C., Carl, G., Carré, G., ... & Lautenbach, S. (2013). Collinearity: a review of methods to deal with it and a simulation study evaluating their performance. Ecography, 36(1), 27-46.

[27] Fang, C., Gao, D., Xue, Y., & Xiong, Z. (2019, July). Research on short-term traffic flow prediction method based on real-time traffic status. In IOP Conference Series: Materials Science and Engineering (Vol. 569, No. 5, p. 052062). IOP Publishing.

[28] G. Stein, B. Chen, A. S. Wu, and K. A. Hua, "Decision tree classifier for network intrusion detection with GA-based feature selection," in Proceedings of the 43rd annual Southeast regional conference-Volume 2, 2005, pp. 136–141.

[29] G. Williams, N. Wagener, B. Goldfain, P. Drews, J.M. Rehg, B. Boots, and E.A. Theodorou, "Information-theoretic mpc for model-based reinforcement learning," In Robotics and Automation (ICRA), IEEE International Conference on, pp. 1714–1721, 2017.

[30] Gow, J., Baumgarten, R., Cairns, P., Colton, S., & Miller, P. (2012). Unsupervised modeling of player style with LDA. IEEE Transactions on Computational Intelligence and AI in Games, 4(3), 152-166.

[31] H. Mehdi, et al., Cloud traffic prediction based on fuzzy arima model with low dependence on historical data, Trans. Emerg. Telecommun. Technol. (2019) e3731.

[32] He, Y., E. Lee, and T. A. Warner. 2017. "A Time Series of Annual Land Use and Land Cover Maps of China from 1982 to 2013 Generated Using AVHRR GIMMS NDVI3g Data." Remote Sensing of Environment 199: 201–217.

[33] I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. U. Khan, "The rise of "big data" on cloud computing: Review and open research issues," Information systems, vol. 47, pp. 98–115, 2015.

[34] I. S. Damanik, A. P. Windarto, A. Wanto, S. R. Andani, and W. Saputra, "Decision Tree Optimization in C4. 5 Algorithm Using Genetic Algorithm," in Journal of Physics: Conference Series, 2019, vol. 1255, no. 1, p. 012012.

[35] I. T. Jolliffe, Principal Component Analysis. New York: SpringerVerlag, 1986.

[36] Inzunza, M. C. H., Robles, L. H., Carlos-Mancilla, M. A., & López-Neri, E. (2020). Traffic Prediction Architecture based on Machine Learning Approach for Smart Cities. Res. Comput. Sci., 149(11), 23-33.

[37] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: a survey," IEEE Transactions on Intelligent Transportation Systems, vol. 12, no. 4, pp. 1624–1639, 2011.

[38] J.T. Wilkes and C.R. Gallistel, "Information theory, memory, prediction, and timing in associative learning," Computational Models of Brain and Behavior, pp. 481–492, 2017.

[39] Jang, B., Kim, M., Harerimana, G., & Kim, J. W. (2019). Q-learning algorithms: A comprehensive classification and applications. IEEE Access, 7, 133653-133667.

[40] Janković, S., Uzelac, A., Zdravković, S., Mladenović, D., Mladenović, S., & Andrijanić, I. (2021). TRAFFIC VOLUMES PREDICTION USING BIG DATA ANALYTICS METHODS. International Journal for Traffic & Transport Engineering, 11(2).

[41] Janković, S., Uzelac, A., Zdravković, S., Mladenović, D., Mladenović, S., & Andrijanić, I. (2021). TRAFFIC VOLUMES PREDICTION USING BIG DATA ANALYTICS METHODS. International Journal for Traffic & Transport Engineering, 11(2).

[42] Jiber, M., Mbarek, A., Yahyaouy, A., Sabri, M. A., & Boumhidi, J. (2020). Road traffic prediction model using extreme learning machine: the case study of Tangier, Morocco. Information, 11(12), 542.

[43] K. Mittal, D. Khanduja, and P. C. Tewari, "An insight into 'Decision Tree Analysis'"," World Wide Journal of Multidisciplinary Research and Development, vol. 3, no. 12, pp. 111–115, 2017.

[44] Kamble, S. J., & Kounte, M. R. (2020). Machine learning approach on traffic congestion monitoring system in internet of vehicles. Procedia Computer Science, 171, 2235-2241.

[45] Kashyap, A. A., Raviraj, S., Devarakonda, A., Nayak K, S. R., KV, S., & Bhat, S. J. (2022). Traffic flow prediction models–A review of deep learning techniques. Cogent Engineering, 9(1), 2010510.

[46] Knapińska, A., Lechowicz, P., & Walkowiak, K. (2021, June). Machine-learning based prediction of multiple types of network traffic. In International Conference on Computational Science (pp. 122-136). Springer, Cham.

[47] Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. Emerging artificial intelligence applications in computer engineering, 160(1), 3-24.

[48] Kumar, R., & Verma, R. (2012). Classification algorithms for data mining: A survey. International Journal of Innovations in Engineering and Technology (IJIET), 1(2), 7-14.

[49] Lee, S., & Fambro, D. B. (1999). Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting. Transportation Research Record: Journal of the Transportation Research Board, 1678(1), 179–188.

[50] Li, J., Boonaert, J., Doniec, A., & Lozenguez, G. (2021). Multi-models machine learning methods for traffic flow estimation from Floating Car Data. Transportation Research Part C: Emerging Technologies, 132, 103389.

[51] Li, Y., & Jiang, W. (2019, November). Research on short-term Traffic flow Prediction Based on Big Data Environment. In 2019 Chinese Automation Congress (CAC) (pp. 1758-1762). IEEE.

[52] Lippi, M.; Bertini, M.; Frasconi, P. 2013. Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning, IEEE Transactions on Intelligent Transportation Systems 14(2):871–882.

[53] Lizotte DJ, Laber EB (2016) Multi-Objective Markov Decision Processes for Data-Driven Decision Support. Journal of Machine Learning Research 17:211:1–211:28.

[54] M. Arslan, A.-M. Roxin, C. Cruz, and D. Ginhac, "A review on applications of big data for disaster management," in 2017 13th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS). IEEE, 2017, pp. 370–375.

[55] M. C. Council. Open data portal of the madrid city council. [Online]. Available: (http://datos.madrid.es).

[56] Magalhaes, R. P., Lettich, F., Macedo, J. A., Nardini, F. M., Perego, R., Renso, C., & Trani, R. (2021). Speed prediction in large and dynamic traffic sensor networks. Information Systems, 98, 101444.

[57] Mahesh, B. (2020). Machine learning algorithms-a review. International Journal of Science and Research (IJSR).[Internet], 9, 381-386.

[58] Makaba, T., Doorsamy, W., & Paul, B. S. (2020). Exploratory framework for analyzing road traffic accident data with validation on Gauteng province data. Cogent Engineering, 7(1), 1834659.

[59] Maxwell, A. E., Warner, T. A., & Fang, F. (2018). Implementation of machine-learning classification in remote sensing: An applied review. International Journal of Remote Sensing, 39(9), 2784-2817.

[60] McLachlan, G. J. (2005). Discriminant analysis and statistical pattern recognition. John Wiley & Sons.

[61] Meena, G., Sharma, D., & Mahrishi, M. (2020, February). Traffic prediction for intelligent transportation system using machine learning. In 2020 3rd International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things (ICETCE) (pp. 145-148). IEEE.

[62] Navarro-Espinoza, A., López-Bonilla, O. R., García-Guerrero, E. E., Tlelo-Cuautle, E., López-Mancilla, D., Hernández-Mejía, C., & Inzunza-González, E. (2022). Traffic Flow Prediction for Smart Traffic Lights Using Machine Learning Algorithms. Technologies, 10(1), 5.

[63] Nellore K, Hancke G (2016) A survey on urban traffic management system using wireless sensor networks. Sensors 16:157.

[64] Nie, L., Ning, Z., Obaidat, M. S., Sadoun, B., Wang, H., Li, S., ... & Wang, G. (2020). A reinforcement learning-based network traffic prediction mechanism in intelligent internet of things. IEEE Transactions on Industrial Informatics, 17(3), 2169-2180.

[65] Nikam, S. S. (2015). A comparative study of classification techniques in data mining algorithms. Oriental Journal of Computer Science and Technology, 8(1), 13-19.

[66] Obulesu, O., Mahendra, M., & ThrilokReddy, M. (2018, July). Machine learning techniques and tools: A survey. In 2018 International Conference on Inventive Research in Computing Applications (ICIRCA) (pp. 605-611). IEEE.

[67] Oladipo, I. D., AbdulRaheem, M., Awotunde, J. B., Bhoi, A. K., Adeniyi, E. A., & Abiodun, M. K. (2022). Machine Learning and Deep Learning Algorithms for Smart Cities: A Start-of-the-Art Review. IoT and IoE Driven Smart Cities, 143-162.

[68] Pal, M. 2005. "Random Forest Classifier for Remote Sensing Classification." International Journal of Remote Sensing. 26 (1): 217–222.

[69] Patel, P., Narmawala, Z., & Thakkar, A. (2019). A survey on intelligent transportation system using internet of things. Emerging Research in Computing, Information, Communication and Applications, 231-240.

[70] Priyanka and D. Kumar, "Decision tree classifier: a detailed survey," International Journal of Information and Decision Sciences, vol. 12, no. 3, pp. 246–269, 2020.

[71] Qu, Z., & Li, J. (2022). Short-term Traffic Flow Forecast on Basis of PCA-Interval Type-2 Fuzzy System. In Journal of Physics: Conference Series (Vol. 2171, No. 1, p. 012051). IOP Publishing.

[72] Qureshi, K.N., & Abdullah, A.H. (2013). A Survey on Intelligent Transportation Systems. Middle-East Journal of Scientific Research, 15, 629-642.

[73] R. L. Villars, C. W. Olofson, and M. Eastwood, "Big data: What it is and why you should care," White paper, IDC, vol. 14, pp. 1–14, 2011.

[74] Rajkumar, S. C. (2019). Optimized traffic flow prediction based on cluster formation and reinforcement learning. International Journal of Communication Systems, e4178.

[75] Ramchandra, N. R., & Rajabhushanam, C. (2021, June). Traffic Prediction System Using Machine Learning Algorithms. In I3CAC 2021: Proceedings of the First International Conference on Computing, Communication and Control System, I3CAC 2021, 7-8 June 2021, Bharath University, Chennai, India (p. 424). European Alliance for Innovation.

[76] Ray, S. (2019, February). A quick review of machine learning algorithms. In 2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon) (pp. 35-39). IEEE.

[77] Resende, P. A. A., & Drummond, A. C. (2018). A survey of random forest-based methods for intrusion detection systems. ACM Computing Surveys (CSUR), 51(3), 1-36.

[78] S. S. Gavankar and S. D. Sawarkar, "Eager decision tree", in 2017 2nd International Conference for Convergence in Technology (I2CT), Mumbai, Apr. 2017, pp. 837–840.

[79] Sayed, S. A., Abdelhamid, Y., & Hefny, H. A. (2023). Traffic Flow Prediction Using Big Data and Geographic Information Systems: A Survey of Data Sources, Frameworks, Challenges, and Opportunities. Int. J. Com. Dig. Sys, 14(1).

[80] Shahraki, A., Taherkordi, A., & Haugen, Ø. (2021). TONTA: Trend-based online network traffic analysis in ad-hoc IoT networks. Computer Networks, 194, 108125.

[81] Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Van Den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M, et al. (2016) Mastering the Game of Go with Deep Neural Networks and Tree Search. Nature 529(7587):484–489.

[82] Singh, G., Al'Aref, S. J., Van Assen, M., Kim, T. S., van Rosendael, A., Kolli, K. K., ... & Min, J. K. (2018). Machine learning in cardiac CT: basic concepts and contemporary data. Journal of Cardiovascular Computed Tomography, 12(3), 192-201.

[83] Steffen, T., & Lichtenberg, G. (2022, February). A Machine Learning Approach to Traffic Flow Prediction using CP Data Tensor Decompositions. In IFAC World Congress 2020. Loughborough Research Repository.

[84] Sun, P., Boukerche, A., & Tao, Y. (2020). SSGRU: A novel hybrid stacked GRU- based traffic volume prediction approach in a road network. Computer Communications, 160, 502–511.

[85] Suneel Kuamr. (2022). Traffic Flow Prediction Using Machine Learning Algorithms. International Research Journal of Engineering and Technology (IRJET), 9(4), 2995–3004.

[86] Świechowski, M., Godlewski, K., Sawicki, B., & Mańdziuk, J. (2021). Monte Carlo tree search: A review of recent modifications and applications. arXiv preprint arXiv:2103.04931.

[87] T. C. M. A. for Planning (CMAP). My daily travel survey. [Online]. Available: (https://www.cmap.illinois.gov/data/transportation/travel-survey#My_Daily_Travel_Survey).

[88] T. Kolajo, O. Daramola, and A. Adebiyi, "Big data stream analysis: a systematic literature review," Journal of Big Data, vol. 6, no. 1, p. 47, 2019.

[89] T. S. D. Center. 2010–2012 california household travel survey. [Online]. Available: (https://www.nrel.gov/transportation/secure-transportation-data/tsdc-california-travel-survey.html).

[90] Taunk, K., De, S., Verma, S., & Swetapadma, A. (2019). A Brief Review of Nearest Neighbor Algorithm for Learning and Classification. 2019 International Conference on Intelligent Computing and Control Systems (ICCS).

[91] Tianchi. Kdd cup 2017 highway tollgates traffic flow prediction dataset. [Online]. Available: (https://tianchi.aliyun.com/dataset/dataDetail?dataId=60).

[92] Tin Kam Ho. 1995. Random decision forests. In 3rd International Conference on Document Analysis and Recognition - Volume 1 (ICDAR'95). IEEE Computer Society, 278–282.

[93] Tin Kam Ho. 1998. The random subspace method for constructing decision forests. 20, 8 (Aug. 1998), 832–844.

[94] Tu, M., Liu, B., & Zhong, F. (2019, December). Research of intelligent traffic flow prediction algorithm. In Journal of Physics: Conference Series (Vol. 1423, No. 1, p. 012038). IOP Publishing.

[95] Upadhyaya, S., & Mehrotra, D. (2022). The Facets of Machine Learning in Lane Change Prediction of Vehicular Traffic Flow. In Proceedings of International Conference on Intelligent Cyber-Physical Systems (pp. 353-365). Springer, Singapore.

[96] Van Der Voort, M., Dougherty, M., & Watson, S. (1996). Combining Kohonen maps with ARIMA time series models to forecast traffic flow. Transportation Research Part C: Emerging Technologies, 4(5), 307–318.

[97] Veropoulos, K., Campbell, C. & Cristianini, N. (1999). Controlling the Sensitivity of Support Vector Machines. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI99).

[98] W. Jiang and J. Luo, "Big data for traffic estimation and prediction: a survey of data and tools," Applied System Innovation, vol. 5, no. 1, p. 23, 2022.

[99] W. Jiang and J. Luo, “Big data for traffic estimation and prediction: a survey of data and tools,” Applied System Innovation, vol. 5, no. 1, p. 23, 2022.

[100] Wang, J., Pradhan, M. R., & Gunasekaran, N. (2022). Machine learning-based human-robot interaction in ITS. Information Processing & Management, 59(1), 102750.

[101] Wang, Z., Chu, R., Zhang, M., Wang, X., & Luan, S. (2020). An improved hybrid highway traffic flow prediction model based on machine learning. Sustainability, 12(20), 8298.

[102] Watkin, C. J. C. H., & Dayan, P. (1992). Technical note Q-learning. Machine Learning, 8(3), 279-292.

[103] Watkins, C.J.C.H. (1989). Learning from delayed rewards. Ph.D. Thesis, University of Cambridge, England.

[104] Weerasekera, R., Sridharan, M., & Ranjitkar, P. (2020). Implications of spatiotemporal data aggregation on short-term traffic prediction using machine learning algorithms. Journal of Advanced Transportation, 2020.

[105] Williams, B. M. (2001). Multivariate vehicular traffic flow prediction: Evaluation of ARIMAX modeling. Transportation Research Record: Journal of the Transportation Research Board, 1776(1), 194–200.

[106] Williams, B. M., & Hoel, L. A. (2003). Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results. Journal of Transportation Engineering, 129(6), 664–672.

[107] World Health Organization. (2018). “GLOBAL STATUS REPORT ON ROAD SAFETY 2018 SUMMARY.” [Online]. Available: https://apps.who.int/iris/bitstream/handle/10665/277370/WHO-NMH-NVI-18.20-eng.pdf?ua=1.

[108] X. Shi, Z. Gao, L. Lausen, H. Wang, D.Y. Yeung, W.K. Wong, and W.C. Woo, "Deep learning for precipitation nowcasting: A benchmark and a new model," Advances in Neural Information Processing Systems," pp. 5622–5632, 2017.

[109] Xiao, J., Xiao, Z., Wang, D., Bai, J., Havyarimana, V., & Zeng, F. (2019). Short-term traffic volume prediction by ensemble learning in concept drifting environments. Knowledge-Based Systems, 164, 213-225.

[110] Y. An, Y. Wang, and H. Meng, "Multi-task deep learning for user intention understanding in speech interaction systems," 2017.

[111] Y. Zhao and Y. Zhang, “Comparison of decision tree methods for finding active objects,” Advances in Space Research, vol. 41, no. 12, pp. 1955–1959, 2008.

[112] Yali Amit and Donald Geman. 1997. Shape quantization and recognition with randomized trees. Neural Computation 9, 7 (1997), 1545–1588.

[113] Yang H, Dillon TS, Chang E et al (2019) Optimized configuration of exponential smoothing and extreme learning machine for traffic flow forecasting[J]. IEEE Trans Ind Inf 15(1):23–34.

[114] Zahid, M., Chen, Y., Jamal, A., & Memon, M. Q. (2020). Short term traffic state prediction via hyperparameter optimization based classifiers. Sensors, 20(3), 685.

[115] Zeroual, A., Harrou, F., & Sun, Y. (2021, December). Predicting road traffic density using a machine learning-driven approach. In 2021 International Conference on Electrical, Computer and Energy Technologies (ICECET) (pp. 1-6). IEEE.

[116] Zheng, L., Yang, J., Chen, L., Sun, D., & Liu, W. (2020). Dynamic spatial-temporal feature optimization with ERI big data for Short-term traffic flow prediction. Neurocomputing, 412, 339-350.

[117] Zou, W., & Xia, Y. (2019). Back propagation bidirectional extreme learning machine for traffic flow time series prediction. Neural Computing and Applications, 31(11), 7401-7414.