A Regression Analysis of Winnipeg Transit On-time Performance in Relation to Traffic Volume

Jason Tran
Department of Computer Science
University of Manitoba
Winnipeg, MB, Canada
tranndt@myumanitoba.ca

Tanisha Turner
Department of Computer Science
University of Manitoba
Winnipeg, MB, Canada
turnert1@myumanitoba.ca

Tommy Wu
Department of Computer Science
University of Manitoba
Winnipeg, MB, Canada
wus2@myumanitoba.ca

Nurida Karimbaeva
Department of Computer Science
University of Manitoba
Winnipeg, MB, Canada
karimban@myumanitoba.ca

Juhee Kim
Department of Computer Science
University of Manitoba
Winnipeg, MB, Canada
kimj4@myumanitoba.ca

***Abstract*—Bus riders desire precision and accuracy when using the Winnipeg transit system. While Winnipeg Transit is responsible for maintaining and delivering public transportation services for the city of Winnipeg, they rely on idealized assumptions regarding real-world bus driving conditions.**
The bus schedule published by the Winnipeg Transit System seems to assume that the buses move at a uniform speed at all times which leads to bus arrival times that are imprecise and inaccurate. Busses in Winnipeg can arrive early, late or on-time. This is obviously not a system that bus riders can rely on, which is especially unfortunate in the frigid winters of Winnipeg.
Given that bus stops cannot have a dynamic schedule, the next best thing would be to create a schedule accounting for the changes in traffic patterns. This is the aim of our project.
We collected and cleaned Lane closures (for construction sites), on-time performance and traffic count datasets, to use as predictive factors for our model.
Using the previously mentioned factors we implemented and tested multiple machine learning algorithms for the best possible fit and prediction. Given the locations of nearly six-thousand bus stops, we created a model to predict whether a bus will arrive early, late, or on-time for any given bus stop for various portions of the day for the City of Winnipeg.

***Index Terms*—Arrival Time, Winnipeg, Bus, Delay, Predictive Analytics, Machine learning, Regression**

I. Introduction

As urbanization continues worldwide, cities are under greater pressure to create efficient public transportation systems. Public transportation management plays a major role in urban development for linking places, creating polycentric urban structure, increasing accessibility, and reducing environmental impacts [2]. Based on these benefits, public transportation is essential for economic growth and labour markets. Therefore, building an efficient public transportation system is necessary in modern cities [2]. Winnipeg is one example of a city of this global trend.

Winnipeg's rapid population growth and increasing city size led to sudden rise in travel demand within the city and a high demand for an efficient public transportation system. The current Winnipeg public transit system has many issues, but we have to chosen to focus on the problem of inaccurate bus arrival schedules. Traditional transportation models do not anticipate delays or early arrivals, which can lead to both economic damages and social impacts for the city and all users of it's public transportation system. Additionally, while there are other studies done on arrival time prediction for transit busses, they are almost all done on large cities with millions of people with well developed and efficient public transportation systems.

Unlike those other cities, Winnipeg is quite small, and while all of its public transit busses are equipped with on-board GPS systems, they also occasionally suffer from transmission problems and issues with their data, that can lead to online methods that rely on that GPS data to be less than accurate. These problems motivated us to research and apply predictive analysis with regards to Winnipeg's transit system to try and find ways of accurately predicting bus arrival times in the city. In order to solve this problem, we will have to adapt to an advanced transport management system that provides accurate and realistic bus arrival times using common delay patterns, which is the main goal of this project.

More precisely, the goal of this project is to predict bus arrival times of Winnipeg's busses, based on several machine learning algorithms, using public data from the city of Winnipeg. *Fig. 1* shows an overview of the bus lines in Winnipeg, which demonstrates the relative size and complexity of Winnipeg's transit system. Each bus line is shown in a different colour, for clarity.

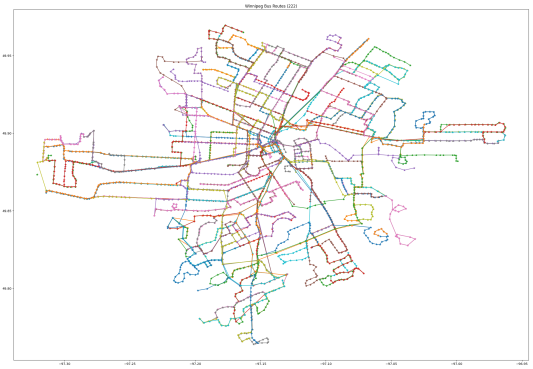

Fig. 1. Overview map of Winnipeg's bus lines

In order to develop our prediction model, we used various experimental variables to begin with. Traffic counts and construction site data were used as factors to estimate the impact on on-time performance but the construction site dataset was excluded since it added little to no improvement to the algorithms' results. This is likely due to the fact that busses may change their route temporarily and some stops could be skipped. The on-time-performance data will have no data on skipped bus stops. Using the training data, we applied different machine learning algorithms and compared their results to find the algorithm that best predicts accurate arrival times.

Our main contribution of this work is summarized as follows:

- We divided training data into two levels (categories), by combining two different original datasets. To be more precise, the bus stops are classified as either a direct traffic count effect zone or an indirect traffic count effect zone, based on the distance between the traffic count location and the bus stop. This extra data cleaning step improved the prediction accuracy. We also ran our tests using predictions from machine learning algorithms train on data from both August 2021 and November 2021, to see if there were any noticeable differences in the accuracy of the predictions from the different months.

II. Related work

A number of bus arrival time prediction models have been developed and have been used on real world applications in our daily life.

The work of Xu et al. [3] gives a good overview of building a precise real-time bus arrival time prediction model based on both real-time information and historical GPS trajectory dataset in a period of two months. Their work [3] shows that the correctness of bus arrival time prediction was improved by using both real-time and historical data from the City of Hangzhou. Traditional bus arrival time prediction methods are mainly based on a single variable, usually on-time performance data(arrival time) [4].

However Zhou et al. [4] propose to consider input dynamic factors that affect bus arrival time, such as weather, road congestion, traffic signal status, construction lanes etc. to improve prediction accuracy. Zhou et al. [4] shows the advantage of using dynamic factors by giving the improved prediction accuracy result with dynamic factors; the dwell time of stations, the passenger number, bus driving efficiency, based on a Recurrent Neural Network in their work based on data from the City of Jinan.

Unlike the studies by Xu et al. [3] or Zhou et al. [4], we used transit data for the City of Winnipeg. We decided on Winnipeg as it is both local to our university, and because of the lack of studies done on bus arrival time prediction in smaller cities, such as Winnipeg.

Our choice in city made our study differ greatly from the others we have mentioned, as Winnipeg is both significantly smaller that Hangzhou or Jinan, and in possession of a much less developed transit system.

While this choice may have caused some issues due to a shortage of thorough data, we believe that it shows that bus time prediction models can be applied to any city, regardless of size, and can be used to provide more accurate transit data to all the city's passengers.

There are a number of methods for bus arrival time prediction models. The existing approaches for the public transportation arrival prediction problem, are the Kalman Filter algorithm, regression models, Support Vector Machines(SVM), Artificial Neural Networks(ANN), and hybrid models [8].

The Kalman filter approach is widely used to predict and estimate future values [8], by using a weighted average [8]. Besides the Kalman filter approach, the regression model is another common approach to predict transport arrival time, although it has limitations due to the strong correlation of the variables of the regression function [8]. Based on the work of Ramkumar et al. [10], Kalman filter performed better than the linear regression algorithm.

The SVM approach is used in bus arrival time prediction extensively [4] and outperforms the other, more traditional, methods [5] [6]. Mingheng et al. [5] explains that Support Vector Machines(SVM) are highly effective in high dimensional spaces, such as urban traffic flow prediction because of the strong nonlinear, stochastic, time-varying characteristics [5] of modern urban transport system.

Although Support Vector Machine has many advantages, Agafonov and Yumaganov [8] suggest using some other effective machine learning algorithms for the public transport arrival time prediction model. In their work [8], they compared the prediction performance of four different bus arrival prediction models: Linear Regression model(LR), Support Vector Regression model(SVR), and two different types of ANN (basic Artificial Neural Network model and extended Artificial Neural Network model) using heterogeneous data input. Agafono and Yumaganov [8] propose an extended ANN model to predict more accurate arrival time compare to LR, SVR and the basic ANN model.

Hashi et al. [9] compared a regular machine learning algorithm

and a hybrid model. Based on the work of Hashi et al. [9], the performance of the hybrid model they proposed, Genetic Algorithm - Support Vector machine with Kalman (GA-SVM-Kalman) is 87.63% better than ANN which yielded 83%.

III. Construction of Data Mining Models

In order to build a prediction model, we followed the steps in the Knowledge Discovery in Databases (KDD) process. In the KDD process the first step is to understand the domain, which in this case, was bus arrival time prediction. We accomplished this by researching the subject using studies such as the studies by Xu et al. [3] and Zhou et al. [4] and then ensuring that we understood the prediction models that they used, and why they used them.

Our next step was to find datasets for our project, for which we used the City of Winnipeg's Open Data Portal [1].

After that we cleaned and preprocessed our data by removing or accounting for all information that was unnecessary, redundant, or incomplete.

The next step in the KDD process would have been to decide what our primary task was, but we had already decided to use our data for prediction, so that was unnecessary.

Then, using our preprocessed dataset, we applied and compared multiple machine learning algorithms in order to achieve the most accurate prediction results.

Finally, we consolidated our results, and used them to create visualizations of our data.

A. Data Collection

The raw data that we used for this study was collected from the City of Winnipeg Open Data Portal. After collecting our data, we divided examined and divided the data into four distinct categories: Road network data, bus stop data, bus arrival time (on-time performance) data, and delay factor data. *(Fig. 2)*

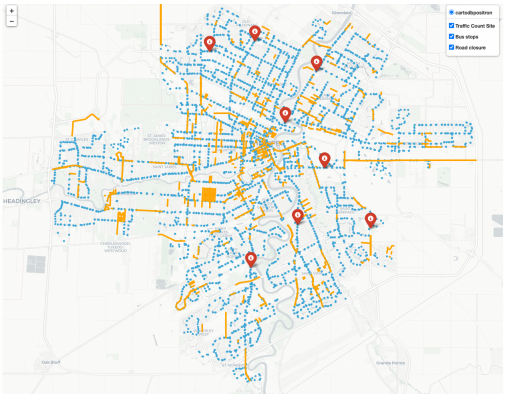

Fig. 2. Overview map of the raw data
Includes the road network, bus stops, traffic count stations, and current lane closures in the City of Winnipeg.

- Road network: Data on the roads in the City of Winnipeg. Includes data for the ID and Block ID, street data such as the street name, type, qualifier, and secondary qualifier, as well as address data, such as the addresses from left, to left, from right, and to right, and other data, such as the number of lanes, the speed limit, speed limit description, whether or not it has reversed geometry and the location. *(Fig. 3)*

Fig. 3. Map of the Road network for the City of Winnipeg

- Bus stops: Data on the bus stops in the City of Winnipeg. The city has about 5155 bus stops in total. Includes data on the stop id, code, name, latitude, and longitude for each bus stop. *(Fig. 4)*

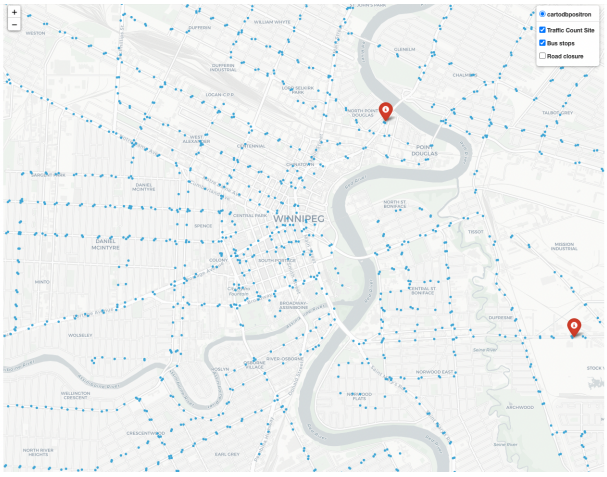

Fig. 4. Map of bus stops in the City of Winnipeg.

- Bus arrival time: Data for on-time performance of busses. Includes stop number, route number, route name, route destination, day type, scheduled time, deviation, and location coordinates. *(Fig. 5)*

[1] https://data.winnipeg.ca

	Row ID	Stop Number	Route Number	Route Name	Route Destination	Day Type	Scheduled Time	Deviation	Location
0	838791619	30893	71	Arlington	Portage via Sinclair	Weekday	2021-08-03T12:57:00	-83	POINT (-97.1487003222228 49.9583003867871)
1	838791621	30894	71	Arlington	Portage via Sinclair	Weekday	2021-08-03T12:57:47	-131	POINT (-97.1447322377209 49.9570676309453)
2	838791623	30884	71	Arlington	Portage via Sinclair	Weekday	2021-08-03T12:58:20	-112	POINT (-97.1459199133062 49.9555484932656)
3	838791626	30377	71	Arlington	Portage via Sinclair	Weekday	2021-08-03T13:00:00	-128	POINT (-97.1452473007769 49.9523715211324)
4	838791628	30378	71	Arlington	Portage via Sinclair	Weekday	2021-08-03T13:01:02	-277	POINT (-97.1409297615775 49.9511780159218)
...	...	...	...	...	...	...	...	...	...
4264663	856899198	40043	47	Transcona - Pembina	Transcona via Regent	Weekday	2021-08-17T10:26:36	-252	POINT (-96.9929264936655 49.8950458350076)
4264664	856899200	40037	47	Transcona - Pembina	Transcona via Regent	Weekday	2021-08-17T10:27:13	-248	POINT (-96.9891662035126 49.8950616196603)
4264665	856899202	40031	47	Transcona - Pembina	Transcona via Regent	Weekday	2021-08-17T10:27:42	-248	POINT (-96.9867783593948 49.8958927013494)
4264666	856899204	40029	47	Transcona - Pembina	Transcona via Regent	Weekday	2021-08-17T10:28:03	-240	POINT (-96.9846475305137 49.8959006485947)
4264667	856899206	40019	47	Transcona - Pembina	Transcona via Regent	Weekday	2021-08-17T10:28:24	-231	POINT (-96.9825558071375 49.8955918271845)

Fig. 5. Raw bus arrival time data

- Delay factors: there were two different delay factors that had an effect on bus arrival times, traffic counts and lane closures.
  - Traffic counts: Data from the permanent count stations in Winnipeg located at McPhillips, Henderson, Disraeli, Pembina, St. Mary's, Inkster, Marion, and Lagimodiere. Includes data on the timestamp, site, side, direction, total amount, longitude, latitude, and location of the traffic at each stop, in fifteen minutes intervals. *(Fig. 6)*

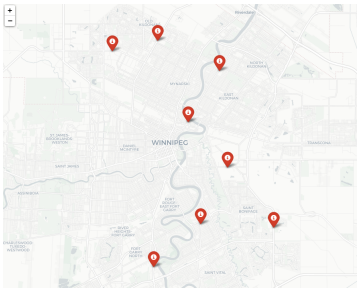

Fig. 6. Traffic count station locations in Winnipeg

  - Lane closure: Data on lane closures in Winnipeg, including the lanes affected, whether the lane was fully, or partially closed, the direction of traffic affected, and the date and time the closure lasted. *(Fig. 7)*

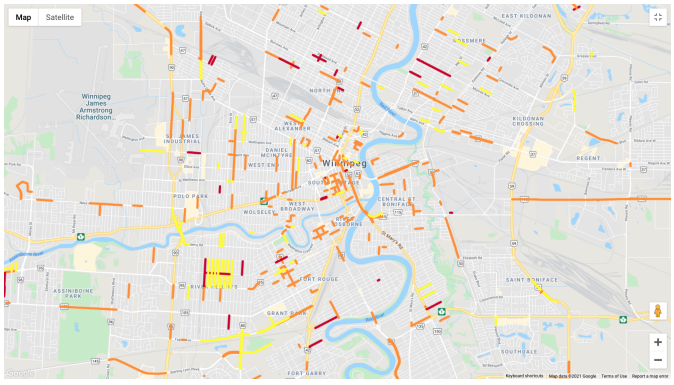

Fig. 7. Map showing lane closures in the City of Winnipeg

- Construction Sites: Originally we attempted to incorporate data on active construction sites into our dataset for our prediction models.
  We gathered information on active construction sites in Winnipeg, such as the location data, including the street, boundaries, and average location, as well as duration data such as the start date, end date, and the times the construction would be active *Fig. 8.*
  Unfortunately, there were complications with the incorporation of the construction data. There was only data available on construction sites that were currently active, and there was no data for any sites that were finished.
  While we did try to apply the construction data regardless, it had little to no impact.
  This could have been because bus stops are connected more by shared routes than by the streets they are located on. For these reasons it was left out of our final preprocessed dataset and our final result calculations.

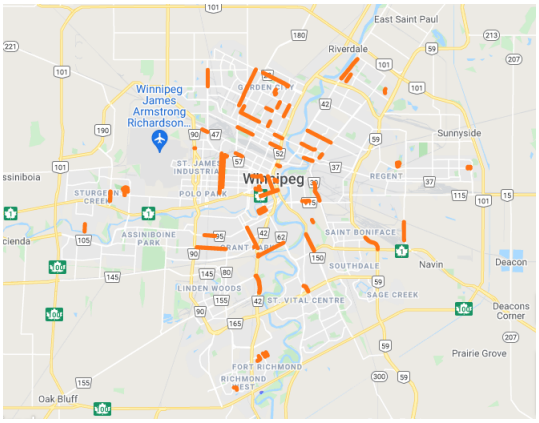

Fig. 8. Map of active construction sites in Winnipeg

B. Data Preprocessing

The raw data that we collected was large and included a great number of redundant features that we needed to exclude before we could start applying data mining algorithms. If we had not cleaned the data before applying the algorithms it would have greatly increased the amount of time needed to process the data.

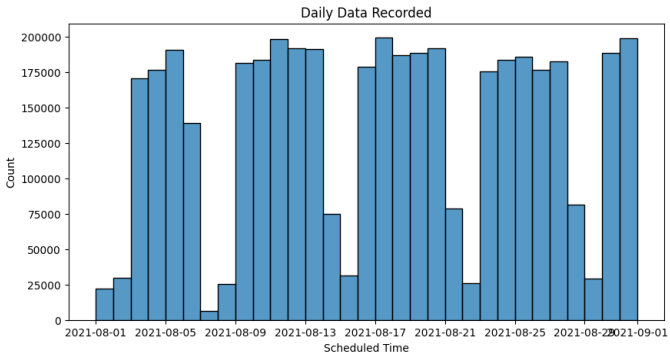

Fig. 9. Graph of Daily On Time Data Recorded

For our road network data we removed the ID and Block Id columns, and extracted street names, the number of lanes the street has, and the location of the street.

From the bus stop data we extracted the street name, stop number, direction (NSEW-bound), latitude, and longitude for each bus stop. Since there are over five-thousand bus stops in the City of Winnipeg, we decided to eliminate all bus stops more that two kilometers from one of the permanent traffic count stations. We did this to ensure that our prediction results would be as accurate as possible, as the traffic count stations can only monitor traffic in their general area. This left us with about fifteen-hundred bus stops in our dataset.

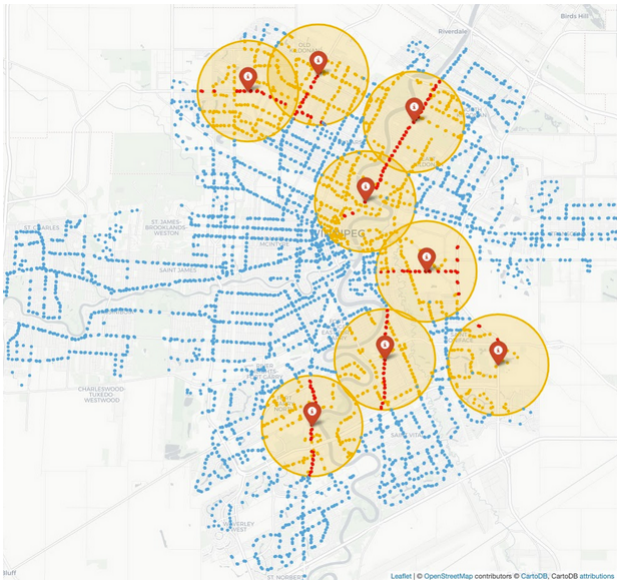

Fig. 10. Map of Winnipeg bus stops and permanent traffic count stations

When cleaning our bus arrival time data we first removed the Row IDs as they were unnecessary for our purposes. Then we converted the Scheduled time column to Date-time format and the Location columns to tuples. Next we grouped the Route Name and Route Destination columns into a single column. Finally we changed the Deviation column from an integer *(Fig. 11)* into a Delay Type column *(Fig. 12)* and categorized all of the delays as follows:

- Early: less than -2 minutes
- On-time: between -2 minutes and 2 minutes
- Short delay: between 2 minutes and 10 minutes
- Medium delay: between 10 minutes and 30 minutes
- Long delay: between 30 minutes and 60 minutes
- Severe delay: more than 60 minutes

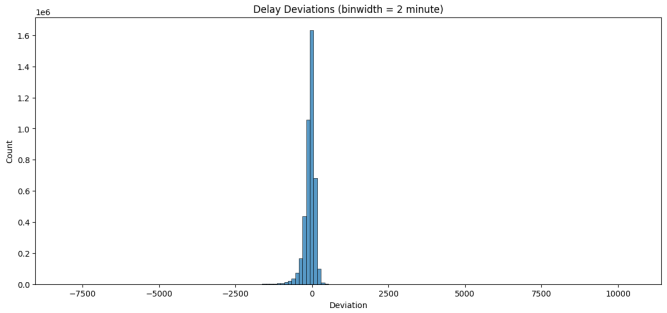

Fig. 11. On-time performance: delay deviations
(Deviation = Arrival Time - Scheduled Time)

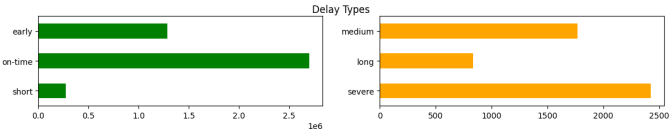

Fig. 12. On-time performance: delay types

Additionally with our arrival time data, we also grouped by days of the week. We did this to account for the fact that weekdays had very different counts that weekends did, as shown in *Fig. 13*.

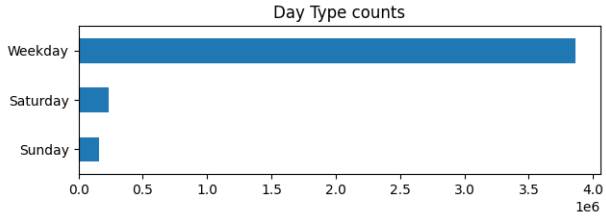

Fig. 13. Bus Arrival Time Data: Day Type Counts

The data preprocessing of our arrival time data helped us narrow down the sheer number of different routes from our dataset *Fit. 14* into a smaller dataset that we could easily combine with others to work with.

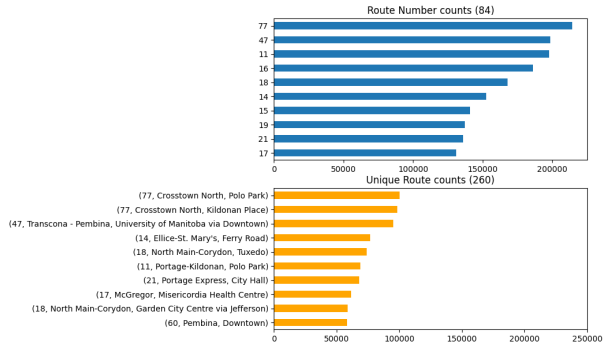

Fig. 14. On-time performance: Route Counts

With our traffic count data, we removed the site, side and location from the data and then extracted the street the station

is located on, the direction and volume of the traffic, the total amount of traffic, and the timestamp for the data. *(Fig.15)*

Timestamp	Northbound	Southbound	Eastbound	Westbound	Total	Street
2021-08-01 00:15:00	14.0	28.0	0.0	0.0	42	St Mary's
2021-08-01 00:30:00	12.0	21.0	0.0	0.0	33	St Mary's
2021-08-01 00:45:00	16.0	18.0	0.0	0.0	34	St Mary's
2021-08-01 01:00:00	10.0	17.0	0.0	0.0	27	St Mary's
2021-08-01 01:15:00	16.0	14.0	0.0	0.0	30	St Mary's

Fig. 15. Sample of preprocessed traffic station data

After cleaning our datasets, we carefully combined them into a single dataset that we could then apply our machine learning algorithms to. Our resulting data included columns for the stop number, street name, traffic count site, distance from the traffic count site, same street, directional, total traffic, average on time performance (OTP), timestamp, time of day, day of week, day of year, time value, and number of lanes as seen in *(Fig.16)*.

	Stop Number	Street	Site	Distance	Same Street	Directional	Total	Arrivals	Average OTP	Timestamp	Time of Day	Day of Week	Day of Year	Time value	Number of Lanes
0	30199	Leila	McPhillips	421.070384	0	0.0	418.0	2.0	-327.0	2021-08-01 02:00:00	2.0	6	213	1627783200000000000	2.0
1	30205	Leila	McPhillips	1689.003285	0	0.0	418.0	2.0	-381.0	2021-08-01 02:00:00	2.0	6	213	1627783200000000000	2.0
2	30206	McGregor	McPhillips	1741.524735	0	0.0	418.0	2.0	-378.5	2021-08-01 02:00:00	2.0	6	213	1627783200000000000	2.0
3	30207	Partridge	McPhillips	1691.791193	0	0.0	418.0	1.0	-28.0	2021-08-01 02:00:00	2.0	6	213	1627783200000000000	2.0
4	30209	McGregor	McPhillips	1718.386291	0	0.0	418.0	1.0	-188.0	2021-08-01 02:00:00	2.0	6	213	1627783200000000000	2.0
5	30211	McGregor	McPhillips	1727.524447	0	0.0	418.0	1.0	-177.0	2021-08-01 02:00:00	2.0	6	213	1627783200000000000	4.0
6	30212	McGregor	McPhillips	1743.271888	0	0.0	418.0	2.0	-328.0	2021-08-01 02:00:00	2.0	6	213	1627783200000000000	4.0
7	30213	McGregor	McPhillips	1744.183619	0	0.0	418.0	1.0	-170.0	2021-08-01 02:00:00	2.0	6	213	1627783200000000000	4.0
8	30214	McGregor	McPhillips	1766.327244	0	0.0	418.0	2.0	-330.0	2021-08-01 02:00:00	2.0	6	213	1627783200000000000	4.0
9	30215	McGregor	McPhillips	1781.773222	0	0.0	418.0	1.0	-159.0	2021-08-01 02:00:00	2.0	6	213	1627783200000000000	4.0
10	30216	McGregor	McPhillips	1815.182063	0	0.0	418.0	2.0	-332.0	2021-08-01 02:00:00	2.0	6	213	1627783200000000000	4.0
11	30217	McGregor	McPhillips	1836.007740	0	0.0	418.0	1.0	-149.0	2021-08-01 02:00:00	2.0	6	213	1627783200000000000	4.0
12	30218	McGregor	McPhillips	1861.761376	0	0.0	418.0	2.0	-333.5	2021-08-01 02:00:00	2.0	6	213	1627783200000000000	4.0
13	30220	McGregor	McPhillips	1912.490434	0	0.0	418.0	1.0	-161.0	2021-08-01 02:00:00	2.0	6	213	1627783200000000000	4.0
14	30221	McGregor	McPhillips	1942.335011	0	0.0	418.0	2.0	-336.0	2021-08-01 02:00:00	2.0	6	213	1627783200000000000	4.0
15	30227	McGregor	McPhillips	1996.575295	0	0.0	418.0	1.0	-151.0	2021-08-01 02:00:00	2.0	6	213	1627783200000000000	4.0
16	30354	Leila	McPhillips	1569.336930	0	0.0	418.0	2.0	-375.0	2021-08-01 02:00:00	2.0	6	213	1627783200000000000	2.0
17	30355	Leila	McPhillips	1307.279722	0	0.0	418.0	2.0	-357.5	2021-08-01 02:00:00	2.0	6	213	1627783200000000000	2.0
18	30356	Leila	McPhillips	1164.119545	0	0.0	418.0	2.0	-347.5	2021-08-01 02:00:00	2.0	6	213	1627783200000000000	2.0
19	30357	Leila	McPhillips	997.106894	0	0.0	418.0	2.0	-337.5	2021-08-01 02:00:00	2.0	6	213	1627783200000000000	2.0

Fig. 16. Sample of combined preprocessed data

C. Data Mining Algorithms

For our data mining algorithms we chose four common machine learning algorithms, that are known to be effective when used as prediction methods, to work with. We chose the Decision Tree Regression, Linear Support Vector Machine, K-Nearest Neighbours Regression, and Random Forest Regression methods as our algorithms.

1) Decision Tree Regression: Decision Tree Regression is a predictive modeling approach that uses a decision tree as a predictive model to go from observations about data (tree branches), to conclusions about data (tree leaves) where the target variable is able to take continuous values. The goal of the created decision tree is to be able to predict the value of some target variable, using the input variables.

2) Linear Support Vector Machine: A Linear Support Vector Machine (SVM) is a predictive modeling approach that uses learning algorithms to analyze and predict data. SVM uses training data to maximize the difference between two categories, and then classifies new data by mapping it in the same way and predicting which category it will belong to based on which side of the gap it lands on.

3) K-Nearest Neighbours Regression: K-Nearest Neighbours Algorithm (k-NN) is a predictive modeling approach where the input consists of the k closest training examples in a training dataset. The output of the algorithm will be the property value for the object, which is the weighted average of the values of the k nearest neighbours, with the weight being the inverse of its distance.

4) Random Forest Regression: Random Forest Regression is a predictive modeling approach that works by constructing many decision trees during training, and the average prediction of the individual decision trees is returned.

IV. Evaluation

Before any of the machine learning (ML) algorithms could be used to make predictions on bus arrival times, they first needed to be trained. The training of the machine learning algorithms was done using supervised learning.

Supervised Learning (SL) in machine learning is the exercise of learning a function that will map an input to an output using labeled training data, made up of training examples of input-output pairs. After being given the training data, the SL algorithm will analyze and generalize the training data, before using it to create a function that can be used to map new examples.

We used randomized test data from our preprocessed dataset for the supervised learning of our machine learning algorithms. We used the months of August 2021 and November 2021 to create the data we used. The time interval was every two hours, as that gave us a large sample of data, without being so large that our machines could not run it. We used bus stops that were within a two kilometer radius of the permanent traffic count stations for our data, as we wanted our results to be as accurate as possible.

We chose test data that we already knew the delays of, so that after training and running the machine learning algorithms, we could then check our results and see which of the machine learning algorithms we chose would provide us with the most accurate predictions for bus arrival times.

A. Data Mining Algorithm Results

After preprocesing our dataset and creating our training dataset, we starting training and testing our chosen machine learning algorithms.

For all of our data mining algorithms were coded to run our algorithms using the functions from scikit-learn [2]. scikit-learn is a free machine learning library that uses the python programming language.

1) Decision Tree Regression: For our decision tree regression algorithm we used the decision tree regression algorithm from scikit-learn [3]. This is a basic decision tree regression algorithm that we used to build a decision tree from input testing data, then we input data that we knew the results of and put the results into an array to be compared to the other algorithms results.

2) Linear Support Vector Machine: For our Linear Support Vector Machine, we used the LinearSVM algorithm from scikit-learn [4]. This is a basic Linear SVM algorithm that we

[2]https://scikit-learn.org/stable/index.html

[3]https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html#sklearn.tree.DecisionTreeRegressor

[4]https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVR.html#sklearn.svm.LinearSVR

gave testing data, then data that we knew the arrival times of, and saved the results into an array to be compared to the other algorithms results.

3) K-Nearest Neighbours Regression: For our K-Nearest Neighbours Regression algorithm, we used the KNeighborsRegressor algorithm from scikit-learn [5]. This is a basic *k*-NN algorithm that we gave testing data, then data that we knew the arrival times of, and saved the results into an array to be compared to the other algorithms results.

4) Random Forest Regression: For our Random Forest Regression algorithm, we used the RandomForestRegressor algorithm from scikit-learn [6]. This is a basic random forest algorithm that we gave testing data, then data that we knew the arrival times of, and saved the results into an array to be compared to the other algorithms results.

B. Testing Results

After running all of the machine learning algorithms and saving the results, we compared the results of the algorithms to determine which algorithm provided the most accurate predictions for bus arrival times.

1) Models' Overall Performance: When comparing overall performance we ran and trained the algorithms with data from both August 2021 and November 2021, and then compares the accuracy with more recent data. We found that the predictions that used the training data from November were the most accurate.
This is likely due to a number of factors such as changes in weather, changes in people riding due to work or school obligations, and changes in public safety regulations due to the pandemic.
Overall, the results of the testing do show a noticeable difference in the accuracy of the algorithms predictions as seen in *Fig. 17* and *Fig. 18*. Since the predictions based on the data from November 2021 are the most accurate, those are the numbers that we will be using in the rest of our tests and comparisons.
Note that all results are means, that were obtained by running the algorithms multiple times and averaging the results.

[5]https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsRegressor.html#sklearn.neighbors.KNeighborsRegressor

[6]https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html?highlight=random%20forest#sklearn.ensemble.RandomForestRegressor

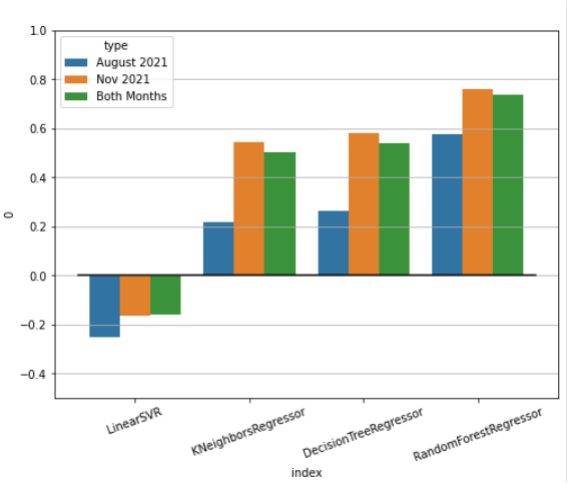

Fig. 17. Model Performance Graph: Overall

R2 score	Aug 2021	Nov 2021	Combined data
LinearSVR	-0.253967	-0.165791	**-0.157980**
KNeighborsRegressor	0.219058	**0.544445**	0.504456
DecisionTreeRegressor	0.264441	**0.582471**	0.540938
RandomForestRegressor	0.574442	**0.759289**	0.737900

Fig. 18. Model performances: Values

2) Time and Stop Data: When comparing the accuracy of the models' predictions, the accuracy generally increased with the thoroughness the of data that we used when determining results, as seen in *Fig. 19* and *Fig. 20*. Removing the time data caused a large decrease in accuracy for all of the prediction methods.
It is also worth noting that removing the stop number had a significant effect on the prediction accuracy of the Decision Tree Regression model, and caused it to be less accurate than the K-Nearest Neighbours models accuracy.

	No Time Data	No Stop Number	All Features
LinearSVR	-1.536164	**-0.157980**	**-0.157980**
KNeighborsRegressor	-0.100097	0.395829	**0.504456**
DecisionTreeRegressor	0.114709	0.217776	**0.542167**
RandomForestRegressor	0.442889	0.527975	**0.737917**

Fig. 19. Model Performance: Time and Stop data values

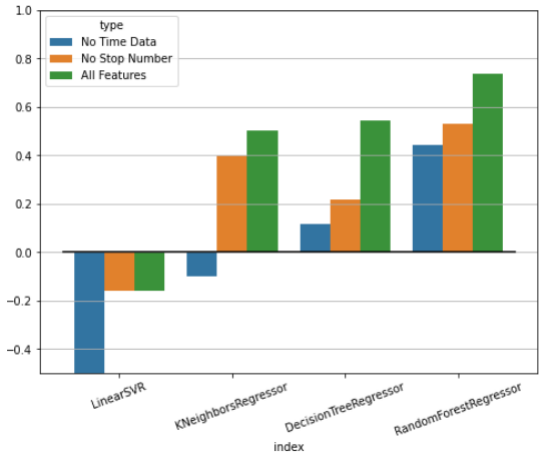

Fig. 20. Model Performance Graph: Time and Stop data

3) Same Street vs Different Street: When comparing the accuracy of the models' predictions, the accuracy generally increased more when we tested with data on the same street, compared to when we tested with data from data that included other streets as shown in *Fig. 21* and *Fig. 22*.

There was a significant increase in the accuracy of all of the non-linear machine learning prediction models when they were trained using data from the same street.

With a powerful enough machine and a larger quantity of data, this could possibly be used to make even more accurate predictions in the future.

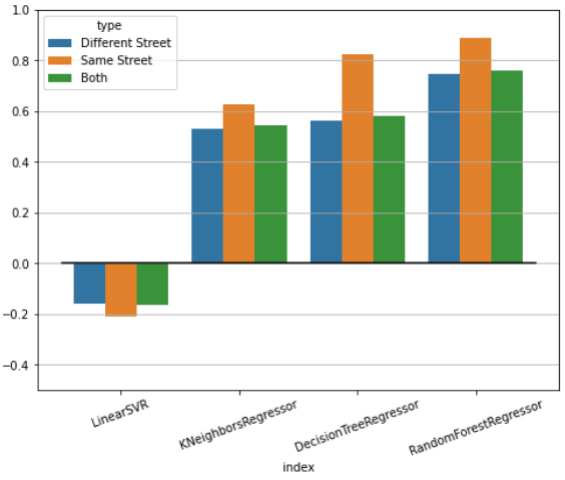

Fig. 21. Model Performance Graph: Same Street vs Different Street

	Different Street	Same Street	Both
LinearSVR	**-0.159971**	-0.210019	-0.165791
KNeighborsRegressor	0.530451	**0.625394**	0.544445
DecisionTreeRegressor	0.564255	**0.822840**	0.582471
RandomForestRegressor	0.746875	**0.887484**	0.759289

Fig. 22. Model Performance: Same Street vs Different Street Values

C. Results with Construction Data

As the results that contained construction data provided little to no increase in accuracy, this was left out of our final computations.

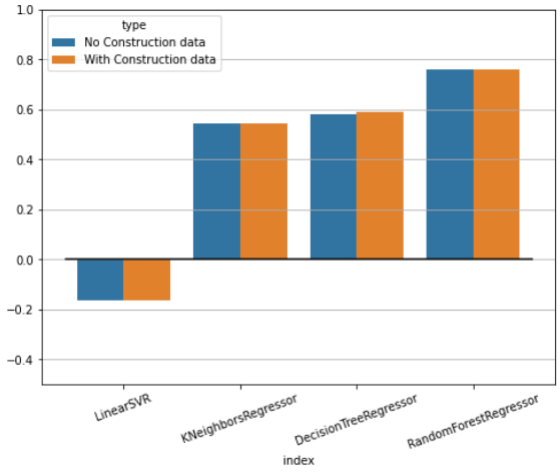

Fig. 23. Model Performance Graph: Construction

	With Construction data	Without Construction data
LinearSVR	-0.165791	-0.165791
KNeighborsRegressor	0.544414	**0.544445**
DecisionTreeRegressor	**0.591031**	0.582471
RandomForestRegressor	**0.761905**	0.759289

Fig. 24. Model Performance: Construction

1) Future Prediction Accuracy: Unfortunately, when we attempted to sue past patterns to predict future results and attempted to apply our machine learning algorithms to more recent data, our results were poor and inconsistent. This could be due to a number of factors, such as the relatively small amount of data that we were able to work with, and the difficulty in applying an average to a specific instance, which can have a larger range of results. *Fig. 25* shows the results of our application of machine learning algorithms. We trained

the algorithm using data from the previous four weeks, then attempted to predict delays for several hours on November 28, 2021.

2021-11-28	12:00	13:00	14:00	15:00	16:00	17:00	...
R2 score	-0.2022	0.1334	-4.1245	-0.6151	-0.3556	0.0258	...

Fig. 25. Sample of prediction data and results from November 28, 2021

V. Conclusion

We found that the key features for performance were the time factors such as the time value, time of Day, Day of the week and day of the year, along with the stop numbers, as they both had a noticeable impact on the accuracy of the machine learning prediction models' results.

We also found from when we ran the machine learning algorithm prediction models that after removing either the time data or the stop data, our results changed significantly. During this test the more data we added, the more accurate our results became. This also showed that removing temporal features from our algorithms, drastically reduced the accuracy of our results. It is worth noting that it may be useful later to try and use a smaller frame of time to see if the results of the algorithms increased in accuracy.

We also found that more specific training data led to more accurate predictions if used in similar locations, as was shown when we trained the algorithms data from the same street, and our accuracy increased greatly. This suggests that with a large volume of similar data, it is possible that we would further increase the accuracy rates of our algorithms.

Overall, the Random Forest Regression algorithm gave the most accurate results, with an average accuracy of 0.759%.

The Decision Tree Regression Models and K-Nearest Neighbours gave similar, but less accurate results with accuracy rates of about 58% and 54% respectively.

However, the Linear Support Vector Machine gave incredibly inaccurate results, which shows that the prediction of bus arrival times is not a linear function.

This implies that if we had access to a large volume of thorough data, that we could possibly get results with a higher accuracy rate.

In this paper, we showed that Random Forest Regression performs better than the other suggested algorithms when it comes to bus arrival time predictions. Additionally, we improved the prediction accuracy by classifying the training data into two categories.

Unfortunately, when we attempted to apply our results to future predictions, they were less than satisfactory. This is most likely due to the limited amount of data that we were able to work with, as well as some other factors.

In conclusion, although the final results of our study were not what we had intended them to be, through completing this project we learned that the data preprocessing step was crucial to the Knowledge Discovery in Databases process, as well processed data noticeably improved both result accuracy and performance speed.

Further, we found that many factors can have a noticeable impact on the accuracy of the results of machine learning algorithms, such as day of the week, the similarity of the training data to the testing data, and the number of factors that are used as inputs for the data mining algorithms.

VI. Limitations

Regrettably, there were a number of limitations that we ran into during our project that had significant impacts on our results.

We encountered issues with our data being too large to properly preprocess and work with, so we could only work with one or two months at a time.

Additionally, removed the construction data from our initial calculations as we only had access to current active construction site data, while we did not have access to past construction data. We also found that when we added what construction data we did have, that it had little to no impact on the accuracy of our results.

As our findings suggest that the more data the model has access to, the better it becomes, the limits of our machines abilities limited us to only being able to preprocess and run a fraction of the total data that could have been available to us, which limited the scope of our dataset, and therefore the accuracy of our results. For an example, data for Transit On Time Performance for the entire year of 2020 was 91 million entries, which were too much for the machines that we had at our disposal to handle.

Additionally, the City of Winnipeg only has eight permanent traffic count stations in the City, which are not enough to provide an accurate image of the current traffic conditions to areas outside of their range.

VII. Future Work

For future work, the more heterogeneous dynamic factors, such as lane closure (due to construction sites), bus route information, weather conditions, traffic signal status, emergency closures, etc. [4] should be considered.

Furthermore, we should take account the implementation of hybrid methods like GA-SVM-Kalman [9] to further improve prediction accuracy. Additionally, since we noted that removing time as a factor substantially reduced the accuracy of our results, in the future, a smaller span of time than the two hour span that we chose could provide more accurate results.

Additionally, we could expand our results further from the permanent traffic count stations than the two kilometers that we chose for this project.

However, that also runs the risk of incorporating data that may not be accurate, as the stations can only monitor traffic in their general area, and are not as accurate from a distance.

We could also, given that Winnipeg has such a varied climate, use data from over a larger period of time, as the both the road conditions of the city and the number of people that use public transit can change seasonally. Those factors could

potentially have impacts on the transit system, and including them in our prediction algorithms could provide us with more accurate results.

In the future we could also take emergencies such as car accidents, floods, and malfunctioning stop lights into account as knowing about them could allow for more accurate arrival times for any busses whose routes are affected.

ACKNOWLEDGMENTS

We would like to express our very great appreciation to our professor Dr. Carson K. Leung for providing us with his incredibly valuable guidance and feedback during the development of this project. It was instrumental in the completion of our project.

We would also like to thank the TAs for our class, *Comp 4710: Introduction to Data Mining* of Fall 2021, at the University of Manitoba: Evan Madill, Adam Pazdor, and Qi Wen for their great assistance to our research.

REFERENCES

[1] Abdul-Rasheed A. Audu, Alfredo Cuzzocrea, Carson K. Leung, Keaton A. MacLeod, Nibrasul I. Ohin, Nadège C. Pulgar-Vidal:
An intelligent predictive analytics system for transportation analytics on open data towards the development of a smart city.
CISIS 2019 (AISC 993): 224-236.

[2] Stjernborg, Vanessa, and Ola Mattisson.
"The Role of Public Transport in Society—A Case Study of General Policy Documents in Sweden."
Sustainability (Basel, Switzerland), vol. 8, no. 11, MDPI AG, 2016, p. 1120–,
https://doi.org/10.3390/su8111120.

[3] Xu, Haitao, and Jing Ying.
"Bus Arrival Time Prediction with Real-Time and Historic Data."
Cluster Computing, vol. 20, no. 4, Springer US, 2017, pp. 3099–106, https://doi.org/10.1007/s10586-017-1006-1.

[4] Zhou, Xin, et al.
"Learning Dynamic Factors to Improve the Accuracy of Bus Arrival Time Prediction via a Recurrent Neural Network."
Future Internet, vol. 11, no. 12, MDPI AG, 2019, p. 247–,
https://doi.org/10.3390/FI11120247.

[5] Mingheng, Zhang, et al.
"Accurate Multisteps Traffic Flow Prediction Based on SVM."
Mathematical Problems in Engineering, vol. 2013, Hindawi Publishing Corporation, 2013, pp. 1–8,
https://doi.org/10.1155/2013/418303.

[6] Bin, Yu, et al. "Bus Arrival Time Prediction Using Support Vector Machines."
Journal of Intelligent Transportation Systems, vol. 10, no. 4, Taylor and Francis Group, 2006, pp. 151–58,
https://doi.org/10.1080/15472450600981009.

[7] Zhang, Xinming, et al.
"An Automatic Real-Time Bus Schedule Redesign Method Based on Bus Arrival Time Prediction."
Advanced Engineering Informatics, vol. 48, Elsevier Ltd, 2021, p. 101295–,
https://doi.org/10.1016/j.aei.2021.101295.

[8] Agafonov, A. .., and A. .. Yumaganov.
"Performance Comparison of Machine Learning Methods in the Bus Arrival Time Prediction Problem."
CEUR Workshop Proceedings, vol. 2416, no. 2416, 2019, pp. 57–62,
https://doi.org/10.18287/1613-0073-2019-2416-57-62.

[9] Hashi, Abdirahman Osman, et al.
"A Robust Hybrid Model Based on Kalman-SVM for Bus Arrival Time Prediction."
Emerging Trends in Intelligent Computing and Informatics, Springer International Publishing, 2019, pp. 511–19,
https://doi.org/10.1007/978-3-030-33582-3_48.

[10] Ramkumar, Neeraj, and Archana Chaudhari.
"Urban Bus Arrival Time Prediction Using Linear Regression and Kalman Filter—A Comparison."
Soft Computing and Signal Processing, Springer Singapore, 2019, pp. 279–87,
https://doi.org/10.1007/978-981-13-3393-4_29.