Towards a Weather-Aware Traffic Model using Deep Belief Networks

A Thesis Proposal
Presented to
the Faculty of the College of Computer Studies
De La Salle University

In Partial Fulfillment
of the Requirements for the Degree of
Bachelor of Science in Computer Science

by

MAGPALE, Nicolle G.
NIEVA, Dyan Raisa L.
REAMON, David Angelo H.
RECCION, Maria Victoria B.

Rafael A. CABREDO, Ph.D.
Adviser

July 18, 2018

Abstract

Numerous traffic models have been developed to alleviate urban traffic congestion, while also considering weather variables such as temperature, wind gust, weather condition, and rainfall. However, while these traffic models do predict traffic congestion and have produced high accuracies, these models were designed and evaluated in the setting of industrialized countries. In countries that have inadequate drainage systems and poor road infrastructure, rainfall does not runoff effectively and causes immense flooding in prolonged rain periods. Hence, flooding has proven to have a substantial impact on road infrastructure as extreme and prolonged rains become a regular occurrence. As such, this research proposes a traffic model that incorporates the aforementioned weather variables and the consequential flooding in Manila. Weather and flood data will be collected and correlated with one another and with traffic condition to identify factors that the model will be trained for. The research will adapt a weather-aware traffic model with the use of Deep Belief Networks and data fusion techniques. Then, the performance of the model with different setups, such as the different fusion levels of the whole model, the fusion techniques for the decision and feature fusion and the individual predictors, will be evaluated using the root mean squared error (RMSE), mean absolute percentage error (MAPE), and sensitivity analysis.

Keywords: Traffic modeling, weather information, deep learning, data fusion

Contents

List of Figures

List of Tables

Chapter 1

Research Description

Several traffic models were already developed to alleviate traffic congestion, while also considering weather as an effect. However, they were not designed and tested in countries with substandard drainage system, low traffic infrastructure investments, and underdeveloped roads. Moreover, these traffic models made use of information on traffic flow represented as continuous data, compared to categorical traffic data. This research aims to adapt a weather-aware traffic model that will be able to predict traffic given categorical data on traffic, and data on weather. This chapter gives a background on the problem of traffic congestion in countries with and without poor drainage management, traffic infrastructure investment, and developed roads, as well as brief overview of the existing state-of-the-art traffic models and their limitations. This chapter also details the research objectives, scope and limitations, and the significance of the research.

1.1 Background of the Study

Traffic congestion has been a natural occurrence in urban centers in past decades (Bando et al., 1995; Zhao et al., 2005) as the number of vehicles steadily increase and outgrow existing road infrastructures, further delaying traffic flow (J. Lee et al., 2015). Aside from traffic signals and accidents, slight commotions (Bando et al., 1995), weather conditions, road accidents, road constructions, and the behavior of drivers (Mahmud et al., 2012) are also considered major causes. Additionally, in developing and newly industrialized countries, traffic congestion is compounded by poor driver discipline, private transit inflation, archaic traffic management, poor road planning, and low traffic infrastructure investment (Jain et al., 2012).

In order to address this problem without curbing urban growth, several countries have developed various traffic models using approaches like a cellular automata-based model, a knowledge-based model, artificial neural networks, and a hybrid model. Inspired by the Biham-Middleton-Levine (BML) model, Hu et al. (2016) developed a cellular automata-based model that maps and predicts traffic jams at intersections in real-time by simulating traffic volume with certain evolution rules. Machine learning techniques were also used to forecast traffic condition. More et al. (2016) used artificial neural networks (ANN) to predict traffic flow using real-time traffic data such as traffic volume, ensuring a smooth flow using a dynamic synchronization of traffic signals. Meanwhile, W.-H. Lee et al. (2009) developed a knowledge-based model that uses location-based services (LBS) to collect data through data mining technologies. Lastly, Pan et al. (2012) created a Historical ARIMA (H-ARIMA) model, a hybrid traffic prediction model that makes use of historical traffic data together with the current traffic data, specifically traffic volume, speed, and occupancy.

These traffic models do indeed predict traffic congestion. However, they did not consider the weather which has a crucial impact on traffic congestion. In fact, adverse weather conditions, such as precipitation, may reduce the visibility of vehicles, loss of stability, and loss of control (Zhao et al., 2005). From all mentioned factors that affect traffic congestion, weather, by itself, is by far the most disruptive, especially in places with poor drainage systems and road infrastructures. In a heavy rainfall, water is not properly managed and usually overflows the roads above the ground. This produces loss of visibility and control of the drivers, causing them to slow down their vehicles in the roads, which, in turn, causes heavy traffic congestion (Varangis et al., 2003).

The effects of weather in predicting traffic congestion have gained the attention of researchers in recent years (Zhao et al., 2005). Thus, different models have been developed to incorporate weather conditions in predicting traffic condition. For instance, Koesdwiady et al. (2016) created a model that incorporates Deep Belief Network (DBN) and data fusion that cross-correlates historical traffic data and current weather data variables making use of a modified ANN to generate a prediction. In Dunne and Ghosh (2013), on the other hand, they created an Auto-Correlated Neural Network (ACNN) taking transformed data using Stationary Wavelet Transform (SWT) as input to produce a prediction. Besides the use of Neural Networks, one model used multiple linear regression (MLR) to analyze traffic congestions using weather data (J. Lee et al., 2015). This model concludes that temperature affects the traffic congestion, while rainfall does not.

As the recent traffic models were designed and evaluated in countries having exceptional drainage system and fair traffic infrastructure investments (e.g. England (M-BML), Ireland (ANN and SWT-ACNN), USA (H-ARIMA and DBN with Data Fusion), Taiwan (Knowledge-based), and South Korea (MLRA)), using them as is might not yield accurate predictions as countries that have substandard drainage system, underdeveloped roads, and low traffic infrastructure investment, such as Manila experience extreme weather conditions on a fairly regular basis and other scenarios not present in other countries. Additionally, the existing studies on traffic modeling made use of data on traffic flow represented as continuous data on traffic volume. Countries that have low traffic infrastructure, insufficient approach on traffic data collection, such as Manila, have data on traffic conditions represented as categorical data. Hence, the analysis of categorical traffic data, and approaches that can be used to extract information on the effects of all weather variables present in the country from such data, was explored.

1.2 Research Objectives

1.2.1 General Objective

The research aims to develop a traffic model that considers data on traffic condition and weather variables such as wind speed, wind gust, temperature, humidity, dew point, precipitation, visibility, pressure, cloud cover, heat index, and feels-like.

1.2.2 Specific Objectives

Specifically, this research aims:

1. To analyze the relationship and effects of weather variables on traffic congestion;
2. To adapt an existing approach on traffic modeling that considers weather variables in predicting traffic congestion; and
3. To test the accuracy and sensitivity of the traffic model.

1.3 Scope and Limitations of the Research

This study used two different publicly available datasets: traffic and weather. The traffic dataset was obtained from the Metro Manila Development Authority

(MMDA). This includes traffic conditions, which were collected in a 15-minute time interval in 14 road segments at Manila for both the northbound (NB) and southbound (SB). On the other hand, the weather dataset was obtained from the World Weather Online (WWO) and generalizes the weather for the entire city of Manila in a 1-hour time interval. This dataset includes weather variables such as wind speed, wind gust, temperature, humidity, dew point, precipitation, visibility, pressure, cloud cover, heat index, and feels-like. The description of each weather variable is shown in Chapter 3. Analysis on traffic and weather were performed to determine the set of features that would be integrated into the model. The traffic and weather datasets were collected from January 2015 to December 2016.

From the approaches and techniques of the existing traffic models that were reviewed (see Chapter 2), this research adapted one weather-aware traffic model, specifically Koesdwiady et al. (2016)'s model using Deep Belief Networks with Data Fusion, based on the similarity of data used and accuracy.

To evaluate the proposed model's performance, two statistical error measures were used, namely, the root mean squared error (RMSE) and mean absolute error (MAE). Furthermore, RMSE was also said to be a good aid for decision making because it describes the enormity of errors (Armstrong & Collopy, 1992). Sensitivity analysis was also performed through the comparison of the model's performance with changing inputs to give insight on the relevance of inputs for the model, and the model's responsiveness to the inputs' changes.

1.4 Significance of the Research

Predicting traffic remains to be a challenging problem in the field of complex systems. On a macroscopic scale, patterns of traffic congestion must be derived from traffic flow, density, and speed (Hueper et al., 2009). However, given that weather also affects traffic, as applied in previous weather-aware traffic models, it is verified that weather factors can be aggregated as a traffic contributing factor. This study would provide further analysis on the trends present on categorical traffic data, and extract information on the effect of weather on traffic. Furthermore, given that weather has an effect on traffic, this study will develop a weather-aware traffic model capable of predicting especially on time periods where weather significantly disrupted traffic, on urban centers with underdeveloped roads and substandard drainage systems.

In transportation, the proposed traffic model could be integrated with existing multi-modal trip planners so that commuters could know how unexpected weather conditions would affect the operations of road-based public transportation. Similarly, for private transits, they could also plan and optimize their trip, and contribute less to the traffic by taking the less congested route.

1.5 Research Methodology

1.5.1 Review of Related Literature

In this phase, existing studies on traffic modeling approaches were reviewed and compared. These approaches could be classified into two: traffic modeling that does not consider weather, and traffic modeling that does consider the weather. To further understand the motivations behind these approaches, and to recognize the factors that build traffic, researches on traffic congestion and the weather's effect on traffic congestion were also reviewed.

1.5.2 Data Collection and Processing

During this phase, historical traffic and weather data were collected. One year of traffic and weather data were collected from publicly available resources. These data were cleaned of missing records and processed to match needs for further analysis.

1.5.3 Data Analysis

To understand the trends and relationships present in both the traffic and weather data, data was analyzed through comparative and correlation analysis. Findings in this phase were used to determine the features to use in the model.

1.5.4 Design and Implementation of Traffic Model

Based on the availability of similar data and accuracy, one traffic model was adapted as the base model for this study. We prioritized traffic models where similar data is accessible for replication before its accuracy. After the replication of the model, it was extended to support other correlated weather factors. These additional factors were integrated into the model either by deriving features out of other traffic models that use it or through experimentation.

1.5.5 Verification and Validation of Traffic Model

In this phase, the performance of the developed traffic model was evaluated using RMSE and MAE. Further on, the performance of different fusion approaches was evaluated to determine what approach is best in predicting traffic. Aside from measuring the performance of the traffic model, it's sensitivity to the diversity of the data, and the variety of its hyperparameters was evaluated to determine the most optimal setting for the model.

1.5.6 Documentation

This phase was done alongside other phases. All findings and developments throughout the research process were documented. The documentation was used to keep track of the progress and implementation of each phase.

Chapter 2

Review of Related Literature

This section surveys two types of traffic models: (a) those that do not consider weather data, and (b) those that do consider weather data. For each traffic model, we compare the data used, their research methodology, the modeling approach, and their evaluation strategy.

2.1 Materials

2.1.1 Traffic Data

The surveyed traffic models used different types of traffic data for training their models. The modified Biham-Middleton-Levine (M-BML) model used real-time traffic density and evolution rules as its input (Hu et al., 2016). On the other hand, an artificial neural network (ANN) model used real-time traffic volume per 5-minute interval as its input (More et al., 2016). Aside from the ANN model, an H-ARIMA model also used traffic volume together with speed data (Pan et al., 2012). These data were collected from 800 loop detectors different stationed at different highways in Los Angeles, California.

On the other hand, the knowledge-based model used vehicle GPS data and intersection delay (W.-H. Lee et al., 2009). This model used location-based services such as GPS to collect data in real time. The data collected were the position of the vehicle (cartesian coordinates), traveling speed, direction, and status (idle or not). In addition, intersection delay, the average time it takes for a vehicle to make its turn in intersections, was also taken into account as an input for this model.

The Deep Belief Network (DBN) with Data Fusion model and the Stationary Wavelet Transform - Autocorrelation Neural Networks (SWT-ACNN) model also collected traffic volume (Koesdwiady et al., 2016; Dunne & Ghosh, 2013). In the DBN with Data Fusion model, traffic volume is measured every 15 minutes for four months to generate a 15-minute traffic prediction. In the SWT-ACNN model, traffic volume is measured hourly in 14 days to generate a 1-hour traffic prediction. The DBN with Data Fusion collected traffic data from freeways, and the SWT + ACNN collected traffic data from city roads. Meanwhile, the MLRA model collected traffic volume together with traffic speed and travel time in road segments in surrounding Ocean Beach in Seoul, South Korea from July and August 2013 (J. Lee et al., 2015).

2.1.2 Weather Data

The weather data used in weather-aware models also differ from one to the other depending on the presence of these variables in their data, and the correlation between weather and traffic. The DBN with Data Fusion collected weather data from 16 different weather stations for 4 months (Koesdwiady et al., 2016). The weather data includes information for different weather variables such as weather condition, temperature, humidity, wind gust and speed, visibility, dew point, and cloud layer height. After correlating weather variables with traffic condition, the DBN with Data Fusion model made use of temperature, wind gust, and weather condition as their final weather input data.

In training the SWT-ACNN model, historical weather data is collected from one weather station (Dunne & Ghosh, 2013). The weather data consisted of hourly rainfall record of rainfall level measured in millimeters per hour during the weekdays in 14 days. However, in the actual run time of the model, current weather condition is collected real-time.

In training the MLRA model, 48 individual weather variables were analyzed through regression. In addition, dummy variables, such as the days of the week, were used. Because there are too many highly-correlated independent variables, regression analysis was used to remove these, avoiding redundancy. Their final input for weather variables is temperature, humidity, and rainfall.

2.2 Research Methodology

The procedure of predicting traffic of the M-BML model can be summarized in three steps (Hu et al., 2016). First, traffic data input is initialized. Based on the traffic density of each road segment in the route, they obtained the average traffic density of each route. Then, the number of vehicles in the M-BML model is initially distributed based on the traffic density of each route.

Second, the M-BML model runs the system. After the model is initialized, a set of evolution rules on how the M-BML model runs is defined: (1) If the next cell is empty, only the vehicles heading *east* can move through one cell at an odd step time; and if the next cell is empty, only the vehicles heading *north* can move through one cell at an even step; and (2) to conserve the number of vehicles in each line, the boundary conditions are periodic. After defining all evolution rules, the system sets a threshold to confirm the jammed area and gradually decreases it to confirm the jammed cells.

Third, different mapping strategies are used to plot the coordinates of the jammed cell in the M-BML model to the intersections in the real-life traffic network. When the model stops running, then the jam value of each cell is confirmed.

Meanwhile, the knowledge-based model undergo four phases as part of its prediction process. The first phase involves the generation of traffic information. This involves the collection and transformation of mined data. This is collected through reports sent to the onboard unit (OBU) of the location-based service application. The mined data will be used to generate real-time traffic information, which is stored and be used as part of the model's real-time predictor's inference.

The next phase involves the mining of the traffic patterns. Using the data from the first phase, traffic patterns will be derived from the historical database. The traffic information generation module will then aggregate these data from the same vehicle and classify them based from four traffic patterns: right-turn delay (RTD), left-turn delay (LTD), spatial and temporal aggregation (STA), and through delay (TD). These processed data will be stored into a journey set table for the model's historical predictor's inference.

After the raw data have been preprocessed and patterns have been mined, an inference engine for the model is developed. This phase will involve the rule construction of the model. These will be based on traffic patterns and meta-rules defined by domain experts. Rules derived from the historical journey set are converted to an if-else ruleset. Meanwhile, meta-rules defined by domain experts include real-time external traffic events such as road accidents.

Lastly, an expert system is developed to make use of these generated and preprocessed traffic information. Three inputs are used in the expert system: an origin-destination pair, start time, and external event module. These are used together with the historical and real-time predictor for the knowledge-based model.

More et al. (2016), on the other hand, have developed a model using ANN approach. For their training, four inputs were taken and the output was predicted. Then, together with the input from the input layer, this predicted output is stored in the context layer which is fed in next iteration for predicting the next one. The process of prediction consists of two subprocesses: feed-forward and backpropagation. During backpropagation, the weights are adjusted depending upon the error.

In the H-ARIMA model, Pan et al. (2012) used traffic parameters like traffic volume and speed to analyze an auto-regression algorithm called Auto-Regressive Integrated Moving Average (ARIMA) and a model that uses average historical data called Historical Average Model (HAM). They were able to infer that because ARIMA uses real-time data, it is only optimal for short-term future prediction while HAM is better to use when doing long-term prediction because it uses the average historical data of a given day of week and time of day. Trying to utilize both models' advantages, they developed a hybrid model that distinguishes whether it is more suitable to use ARIMA or HAM given a situation. In training the model, it first initializes the dataset of all historical data on a given day at a given time. This dataset is then used to test both models to determine their prediction error. From there, the model can determine which submodel to use.

In Dunne and Ghosh (2013), the SWT-ACNN model decomposes input data, either traffic or weather or both, using Stationary Wavelet Transform (SWT) into approximations, and feeds these into the ANN, and generates an output which is recombined using Inverse SWT (ISWT). This generated and recombined output is the predicted traffic flow. Before feeding the decomposed data into the neural network, the decomposed data is auto-correlated to determine the correlation between observations at different lags and to determine the most influential point of the approximations. Additionally, the coefficients for both the weather data and traffic flow are used as input to the neural network at each level of decomposition.

The model framework of the SWT-ACNN is comprised of two parts, the Dry Model and the Wet Model Dunne and Ghosh (2013). The model first determines whether is it currently raining. If it is, the Wet Model, which involves the weather data in decomposition and prediction, is activated. Otherwise, the Dry Model which only involves the traffic data in decomposition and prediction is activated. The correlation, generated by auto-correlation and the neural network, between traffic flow and weather indicates the effect of the latter to traffic

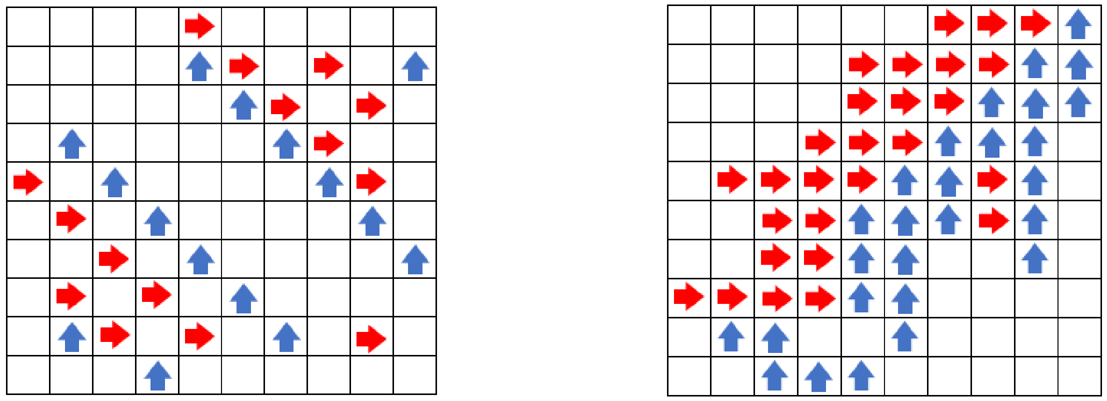
In Koesdwiady et al. (2016), the DBN with Data Fusion model first determines weather variables that truly affect traffic by cross-correlation the different weather variables to the traffic flow. After determining the most influential weather variables, these factors and the traffic flow data is fed into the DBN for training. Traffic and weather are predicted separately. These predictions are fused using Data Fusion techniques to generate an enhanced and accurate traffic flow prediction.

Traffic flow data and weather data used by the DBN with Data Fusion are pre-processed to align data with the model Koesdwiady et al. (2016). Traffic flow data was originally sampled every 30 seconds until aggregating the data into 15 minutes. On the other hand, the weather data was originally sampled every 1 hour. Linear Interpolation was used to resample the weather into 15-minute data. There were observable fluctuations and patterns of the traffic flow especially the differences in traffic during weekdays and weekends. Thus, the data was preprocessed into a detrended version, and a weekday/weekend version. Detrending was used to remove fluctuations caused by variation of the hours and days of each week. The research conducted observed accuracy between the original, detrended and weekday/weekend version.

J. Lee et al. (2015), on the other hand, suggested a model that uses multiple linear regression analysis (MLRA) model. Before everything else, the weather data was cleaned. Because there are too many independent variables in the model that are highly correlated with one another, regression analysis was used to remove these variables. This method of removing variables comprises of three steps. First, this step is called *Backward*, where the model is simplified by removing unnecessary variables one by one. After removing 18 variables, there remain unnecessary independent variables. Hence, their multicollinearity will be diagnosed to remove them again. Second, this step is called *MultiCollinearity*, where the independent variables are removed again based on their multicollinearity (level of linear relationship) between them. 18 more variables were removed after this step. Third, this step is called *Significant Probability (p-value)*, wherein the remaining weather variables are thoroughly filtered.

2.3 Modeling Approach

Different modeling approaches have been used by the models. The M-BML model was inspired by Biham, Middleton, and Levine's model called the BML (Hu et al., 2016). The latter model was the first two-dimensional cellular automaton (CA) model ever developed for simulating traffic in urban road networks (Biham et al., 1992). Each lattice in the two streets, northwards and eastwards. To simulate movement, eastward vehicles move one unit at each odd step time, while northward vehicles move one unit at each even step time, shown in Figure 2.1a. A vehicle cannot move if the next square lattice is occupied by another vehicle. Thus, a traffic jam can be simulated, as shown in Figure 2.1b.



(a) Free Flow State

(b) Jammed State

Figure 2.1: Two states in the BML model. The red arrows are for eastward vehicles, while the blue arrows denote northward vehicles.

However, Hu et al. (2016) modified the BML in such a way that it improved its performance. While the BML distributed the number of vehicles in eastwards and northwards equally, the M-BML distributed at random to be more realistic. In addition, current urban networks can be mapped immediately into the M-BML model for predicting traffic congestion.

W.-H. Lee et al. (2009), meanwhile, developed a knowledge-based real-time travel time prediction model that uses location-based services (LBS) collected through data mining technologies. These LBS are collected from vehicles equipped with GPS and mobile data communication module to report its current speed, traveling direction and its current position. It also uses intersection delay patterns, discovered through sequential pattern mining, which is used to estimate the time to turn to an intersection. The model makes use of a dynamic linear combination of two weight predictors: the historical predictor and real-time predictor. Historical predictors are based on historical traffic information. It is used to estimate link travel time and predict intersection delays. On the other hand, the current predictor is based on real-time traffic information. It is used to incorporate external events into the travel time prediction of the model. The concept of dynamic linear combination allows the model to shift between these predictors. For instance, in the case of an external event (e.g. car accident), the model will reduce the weight of the historical predictor because the current predictor may be more reliable since the effect of this incident will immediately affect the traffic condition.

The H-ARIMA model, on the other hand, makes use of 2 different submodels: ARIMA and HAM (Pan et al., 2012). The ARIMA is an autoregressive moving average model that heavily relies on the combination of previous data collected just before the current time in determining the traffic condition for the next time series. Because it considers recent instances, the ARIMA is more optimal to use for short-term prediction. The HAM, on the other hand, relies more on the average of the previous data given that the data is on the same day of the week and at the same time of the day. In order to distinguish the most suitable approach for a given situation, The researchers trained a decision tree that selects which approach to use. Once given an input, the H-ARIMA feeds the input to both approaches and then gets the overall rate of the prediction error of both approaches. The overall rate of the prediction error is computed as the prediction error of the ARIMA divided by the sum of prediction errors of both approaches. If for instance, the rate of the ARIMA is less than the set threshold (0.5), it means that the ARIMA is to be used for the given situation.

Saputri and Lee (2013) believed, however, that using ANN is the best method of forecasting amongst all others. Inspired by the nonlinear characteristics of Biological Brain System, the work performance of ANN can be compared to the workings of the human brain system. Besides its simple computation and fast performance, More et al. (2016) concluded that ANNs minimize the error in limited time; hence, it improves the efficiency and accuracy of the system.

There were two models that used ANNs in their approach. One of them developed a traffic model that uses Jordan's Sequential Neural Network, which has a good ability of generalization (More et al., 2016). The structure of this kind of ANN is such that the distribution of nodes in the hidden layer should be the square of the number of nodes in the input layer. For example, if there are 5 nodes found in the input layer, then there must be 25 nodes in the hidden layer. The Jordan's Neural Network contains four layers: (a) context layer, which acts as a memory and stores previous information, (b) input layer, which constitutes the input for the next processing, (c) hidden layer, which gets input from the "true" input layer and the context layer, and (d) output layer, which outputs the result or feedbacks to the context layer to be processed again. Figure 2.2 shows a Jordan's Sequential Neural Network with 5 inputs in the input layer.

The ANN algorithm in the SWT + ACNN model, on the other hand, uses the structure of FeedForward (FF) Back Propagation (BP) algorithm for training (Dunne & Ghosh, 2013). The FF phase receives the input and passes it to the output layer through the hidden layer. As discussed earlier, the input fed into the

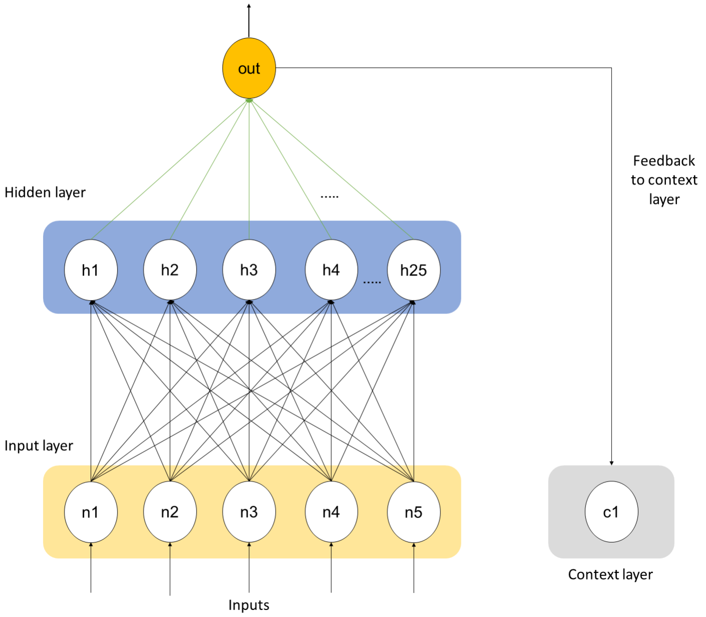


Figure 2.2: ANN Framework used in More et al., 2016

neural network of the SWT + ACNN model is generated from auto-correlation. The output values are calculated from the input layer passing through the hidden layer, and is summed with the weights of the neural network units through activation functions. The BP phase compares the generated to the desired value. This phase optimizes the error function through a number of iterations until the error function is close to optimal. The ANN used by the SWT + ACNN, which takes in auto-correlated inputs, have been labeled as the Auto-Correlation Neural Network (ACNN).

In the developed model of Koesdwiady et al. (2016), Deep Belief Network (DBN) and data fusion techniques are used to enhance the generated prediction. DBN, unlike ANN, is composed of multiple layers of hidden units. The DBN of this model makes use of Restricted Boltzmann Machines (RBM), and stacks RBM together within the hidden layer. RBM is an unsupervised learning algorithm that can learn useful features of the data. It takes the input and translates them into a set of numbers that represents them. Then, these numbers can be translated back to reconstruct the inputs. This neural network is trained through multi-task learning to predict several traffic flow predictions at each level at the same time to reach the final traffic flow prediction. Koesdwiady et al. (2016) further discusses that the optimal architecture for the DBN in traffic prediction consists of three hidden layers with 250 in the first, 200 in the second, and 100 in the last hidden layer. Additionally, Koesdwiady et al. (2016) adds that 100 epochs is optimal for training the DBN. The separately predicted traffic flow and weather data will be fused using data fusion techniques to generate an enhanced and accurate traffic flow prediction. The DBN with Data Fusion makes use of Decision In - Decision Out (DEI-DEO) which fuses input data considered as the decision to generate a new or enhanced decision (Koesdwiady et al., 2016).

J. Lee et al. (2015), on the contrary, suggested a traffic model that makes use of multiple linear regression analysis (MLRA) employing both weather forecast data and traffic congestion data. MLR analysis is used to explain the relationship between one continuous dependent variable (traffic congestion data) and two or more independent variables (weather data). In this model, the traffic congestion is influenced by 48 independent variables. However, in their research, some of those independent variables are highly correlated to each other, and therefore should be removed to prevent the increase of correlation.

2.4 Evaluation Strategy

Before computing the accuracy of each traffic models, different experiments and case studies were first performed. All traffic models were tested to predict the traffic congestion at a given time period.

In evaluating the knowledge-based model, (W.-H. Lee et al., 2009) collected five months of location-based service (LBS) raw data. The LBS raw data was collected through an online taxi dispatch system (TDS), consisting of around 500 taxis operating 24-7 in a Taipei urban area. The first four months are used for mining traffic patterns, whereas the fifth is used for verifying the accuracy of the model. In the experiment, a random origin-destination pair is selected at two peak hour sections. The model produced a precision of 10.8% *relative mean error* (RME) and *root mean squared error* RMSE of 15.92%. To further challenge its reliability, the experiment was performed with the same set of input, having the intersection delay replaced with a turning delay estimate from a human expert. The model produced a precision of 20% RME and 30.55% RMSE, which, in comparison with the pattern generated intersection delay, is relatively unreliable.

In evaluating the accuracy of the H-ARIMA model, the researchers compared results with other baseline approaches given two situations: short-term prediction and long-term prediction (Pan et al., 2012). RMSE and *mean absolute percentage error* (MAPE) were used to measure the accuracy of the traffic prediction. Although the H-ARIMA model was able to yield better results than the other baseline approaches for short-term prediction, the researchers were not able to see the edge of the H-ARIMA model against them. Similar to that, the H-ARIMA model was also able to produce a better result for long-term prediction compared to the baseline approaches. It was observed that its MAPE is lower compared to others because it considers historical traffic data in predicting traffic. Overall, the H-ARIMA model was able to produce around 95-96% accuracy.

To measure the accuracy of the predictions of the DBN with Data Fusion, performance indexes RMSE and MAE are used (Koesdwiady et al., 2016). These performance indices calculate the error value of the prediction by comparing the actual and predicted output at a certain time t. The research compared the performance of different neural networks of ARIMA and ANN with DBN. After experimentation, the DBN outperformed the other neural networks in predicting traffic flow. The performance of the DBN generated an average MAE of 0.07 and average RMSE of 0.05. On the other hand, ANN generated an average MAE of 0.08, and an average of RMSE of 0.06, and ARIMA generated an average MAE of 0.27 and an average RMSE of 0.23.

To examine the performance of the neural network with low, medium and heavy traffic, three different freeways that experience the mentioned traffic conditions are chosen (Koesdwiady et al., 2016). The detrended version of the data was more accurate in predicting heavy traffic, and the original data was more accurate in predicting low and medium traffic. Medium traffic flow using original data achieved an MAE of 0.041 and RMSE of 0.06. Low traffic flow using original data achieved MAE of 0.034 and RMSE of 0.045.On the other hand, detrended version of the data performed best in predicting high traffic flow achieving an MAE of 0.06 and RMSE of 0.09. Moreover, using weekday/weekend version of the data generated a higher average error. Furthermore, the model better predicts the medium traffic flow with consideration to weather information.

The model's performance was compared to ANN that takes in undecomposed data to observe the differences in performance. To measure the accuracy and performance, RMSE and MAPE were used. The SWT-ACNN outperformed the non-wavelet model significantly by approximately 5% of MAPE, and 40 of RMSE. Additionally, the wet model is superior to the dry model achieving a MAPE of more than 30% than the dry model.

To compare the results of J. Lee et al. (2015)'s MLRA model with the actual values, present traffic congestion data were used. As discussed in the previous section, the weather data used in their research are temperature, humidity, rainfall. In their analysis, they used MAPE to assess the developed MLRA model. When they predicted the traffic congestion from July and August 2013, the traffic model got an accuracy of 94.1%. However, when they predicted the traffic congestion in 2014 for the same months, they have arrived at 84.8%.

Table 2.1: Summary of Traffic Models

Author (Year)	Modeling Approach	Traffic Data Input	Weather Data Input	Coverage	Evaluation Strategy	Accuracy
Hu et al. (2016)	Cellular Automata (CA)	- traffic density - evolution rules	none	- road segment - Birmingham, England	not indicated	94.00%
More et al. (2016)	Artificial Neural Network (ANN)	traffic volume	none	- road segment - Dublin, Ireland	not indicated	92.00 - 98.00%
W.-H. Lee et al. (2009)	Knowledge-based	- vehicle GPS data - intersection delay	none	- whole city - Taipei, Taiwan	- RMSE - RME	84.08%
(Pan et al., 2012)	HAM and ARIMA (H-ARIMA)	- traffic volume - speed	none	- road segment - Los Angeles, USA	- RMSE - MAPE	95.99%
Koesdwiady et al. (2016)	Deep Belief Network with Data Fusion (DBN with Data Fusion)	- traffic volume	- temperature - wind gust - weather condition	- road segment - San Francisco, USA	- RMSE - MAE	95.95%
Dunne and Ghosh (2013)	Stationary Wavelet Transform and Auto-Correlation Neural Networks (SWT-ACNN)	- traffic volume	- rainfall	- road segment - Dublin, Ireland	- RMSE - MAPE	91.45%
J. Lee et al. (2015)	Multiple Linear Regression Analysis (MLRA)	- speed - travel time - traffic volume	- temperature	- whole city Seoul, South Korea	- MAPE	84.80%

Chapter 3

Theoretical Framework

3.1 Traffic-related terms

Traffic flow

According to May (1990), traffic flow studies how travelers interact with infrastructures and with each other, in order to understand the movement of traffic and how can it be improved with less congestion. These travelers include drivers, pedestrians, among others, while infrastructures include roads, signages, stop lights, and other devices that control traffic.

Traffic speed

The speed of a vehicle can be computed by measuring the distance it traveled per unit time. Because the speed of each vehicle may be different from its neighboring vehicles, traffic speed is the average of the speed of each vehicle in the given environment.

Traffic volume

Traffic volume refers to the number of vehicles present in a certain road at a given time.

Traffic density

Given a length of a road, traffic density is measured by the number of vehicles present. This means that if the traffic density is low, then there is no traffic congestion because the vehicles are apart from each other. In reverse, if the traffic density is high, then the vehicles are close to each other; thus, a traffic congestion.

Traffic condition

Traffic conditions refer to the states of traffic at a certain point at a certain time, with respect to traffic flow, speed, volume, and density. It has three major states: light, moderate, and heavy traffic.

3.1.1 Traffic Congestion

Traffic congestion, according to (Vuchic et al., 1994), happens when traffic volume on a particular road goes beyond the road's capacity. (Bovy et al., 2002) define traffic congestion as the state of traffic flow on a particular road with low speeds and high densities compared to some other state with high speeds and low densities.

Road and road segments

A road is a wide path from one place to another, usually paved for the vehicles to drive on (Wang et al., 2013). Typically, a road is created for vehicles to head two opposite directions, northbound and southbound. A road segment, on the other hand, is a portion of a road heading in one direction, separated by infrastructures (e.g. train stations, landmarks, intersections).

3.2 Weather-related terms

Temperature

Temperature is the measure of the hotness and the coldness of an object. There are three major scales that are being used internationally, *Celsius*, *Fahrenheit*, and *Kelvin*.

Dew point

Dew point is the temperature wherein moisture or *dew* appears on solid surfaces. This is caused by condensation, in which the water vaporizes into its liquid form at certain amounts of pressure.

Humidity

Humidity refers to the quantity of the water vapor contained in the air. Measured in percentage, relative humidity, on the other hand, is the quantity of water vapor in relation to how much the air can hold at a certain temperature. If the air is warm, it has high relative humidity since the air holds more water vapor than cold air.

Wind speed

Wind speed is the average wind speed in the indicated amount of time. Because of the changing temperature, wind speed results from the air going from above average temperature to a lower temperature.

Wind Gust

Wind gust is the short, rapid rise of wind speed before quickly calming. The wind is forced to change its direction and speed quickly because of sudden decrease in temperature, wind shears, and friction.

Precipitation

Precipitation refers to any result of air condensation that falls due to the pull of gravity. This includes, but not limited to, rain, snow, and hail. Usually, precipitation is measured in terms of millimeters (mm).

Visibility

Visibility is the measure of how long can we distinguish light or an object at a certain distance.

Pressure

Pressure, also called atmospheric pressure, is the amount of force exerted by the air molecules in a particular surface area of the earth. Meteorologists use millibars (mb) to measure pressure.

Cloud Cover

Cloud cover is defined as the amount of clouds in a fragment of the sky in a particular location. This contributes to the weather condition and visibility. This is usually measured by *okta*.

Heat Index

Heat index is the temperature of how hot the human body feels like. This is directly related to air temperature and relative humidity.

Feels-like

Feels-like is the temperature of how cold the human body feels like when the wind touches the skin.

3.3 Historical Data

Historical data is an extensive archive of data that can be utilized to detect regularities or pattern for predicting future outcomes. It ranges from months-worth to years-worth of past data. The following section features sources whose historical data will be used for the research.

3.3.1 Traffic Data from MMDA's Traffic Monitoring System

The traffic dataset includes traffic conditions for nine major roads in Metro Manila, collected in a 15 minute time interval. The traffic condition can either be *light* (L), *moderate* (M), or *heavy* (H). However, if no information is available, it can also appear as *none* (N).

Major roads from this dataset include EDSA, Commonwealth, Quezon Avenue, España, C5, Ortigas, Marcos Highway, Roxas Boulevard, and SLEX. Each road consists of a number road segments, which includes a northbound lane and a southbound lane. For instance, EDSA consists of 37 road segments (e.g. Balintawak, Kaingin Road, Muñoz).

Table 3.1 shows a sample raw data collected from MMDA's traffic monitoring system. From the sample data, it can be observed that despite not having a traffic condition entry, the interval is still being recorded as part of the dataset, having a traffic condition record of *none* (N).

Table 3.1: Sample of Traffic Data from MMDA's Traffic Monitoring System

Date and Time	Road	Segment	Condition (Northbound)	Condition (Southbound)
2015-01-01 00:00:00	EDSA	Quezon Ave.	L	L
2015-01-01 00:00:00	EDSA	Taft Ave.	M	M
2015-01-01 00:00:00	ESPAÑA	Welcome Rotunda	L	L
2015-01-01 00:15:00	EDSA	Quezon Ave.	N	N
2015-01-01 00:15:00	EDSA	Taft Ave.	M	M
2015-01-01 00:15:00	ESPAÑA	Welcome Rotunda	L	L

3.3.2 Weather Data from World Weather Online

The weather dataset from World Weather Online includes temperature (in both Celsius and Fahrenheit), humidity (in percentage), pressure (in millibars), wind speed (in both miles per hour and kilometers per hour), and dew point (in both Celsius and Fahrenheit), cloud cover amount (in percentage), heat index (in both Celsius and Fahrenheit), visibility (in kilometers), wind chill temperature (in both Celsius and Fahrenheit), wind gust (in both miles per hour and kilometers per hour), feels-like temperature (in both Celsius and Fahrenheit), and precipitation (in millimeters). This data is sampled every one hour of every day and is a generalized reading for the whole city of Manila.

Table 3.2 shows a sample raw data collected from World Weather Online. It could be observed that the conversion of some variables to different units has already been performed as part of the data. Furthermore, missing data is not evident from the samples collected.

Table 3.2: Sample of Weather Data from World Weather Online

Property	Sample Value
Date	2015-10-01
Time	0
Weather Condition	Moderate or heavy rain shower
Temperature (°C)	27
Temperature (°F)	81
Wind Speed (m/h)	2
Wind Speed (km/h)	4
Humidity (%)	87
Visibility (km)	8
Pressure (mb)	1013
Cloud Cover (%)	27
Heat Index (°C)	31
Heat Index (°F)	88
Dew Point (°C)	25
Dew Point (°F)	76
Wind Chill (°C)	27
Wind Chill (°F)	81
Wind Gust (m/h)	4
Wind Gust (km/h)	7
Feels Like (°C)	31
Feels Like (°F)	88
Precipitation (mm)	2.8

3.4 Working Day

A working day refers to a day in which people are assigned on duty in an organization (e.g. company, government, school, etc.) each week (J. Liu, Zhou, & Chen, 2008). For most organizations, working day is defined to be from Mondays to Fridays. Non-working days, meanwhile, are from Saturdays to Sundays. As people are on duty during working days, this implies an increase in demand on transportation when going to their respective organizations (Z. Liu et al., 2016).

Thus, there is an increase in traffic volume due to the demand during weekdays as compared to weekends.

Aside from weekends, there are also instances in which a weekday can be classified as a non-working day. This occurs during holidays or whenever a public-sector or government announces class or work suspension.

3.5 Peak Hour

A peak hour, or rush hour, is a time of the day where the congestion of traffic in roads or inflation of people in public transports are at its highest peak (Downs, 2005). There are two peak hours each weekday: the morning peak hour and the afternoon or evening peak hour. These are the times when majority of the people travel and commute to go to work or school. The morning peak hours depict the time when employees and students travel towards their workplace or school, while the evening peak hours suggest the time when they return to their respective homes. These periods may last for more than one hour, and may vary from road to road, city to city, country to country, and seasonally.

3.6 Extreme Weather Disruptions

Extreme weather disruptions such as typhoons and low-pressure areas have brought heavy and prolonged precipitation which often causes flooding in some areas. One example of these extreme weather disruptions is Typhoon Goring that occurred from July 22, 2015 until July 26, 2015. Weather disruptions typically produce a significant amount of precipitation in just one day. These prolonged rain periods usually last for days, thus building up the effects it brings as time goes by.

3.7 Traffic Model

Mathematically, a traffic model is a representation of traffic in the real-world (Mahmud & Town, 2016). These models exist to help researchers understand how traffic works and how can the congestions be minimized by simulating them in a way that researchers can explore and manipulate.

There are two major types of traffic modeling: *macroscopic* and *microscopic*

(Mahmud & Town, 2016). A macroscopic traffic model deals with the properties of transportation elements in the bigger picture. This includes the study of traffic flow, density, speed, volume, among others. A microscopic traffic model, on the other hand, deals with characteristics of the individual transportation elements. This covers driver behavior, how the vehicles move, among others.

3.8 Linear Interpolation

Using surrounding data points, interpolation can be used to determine the value of an unknown point in a line. Linear interpolation is one method of interpolation where it assumes a linear or straight line relationship between two known points. It is used to approximate value through the weighted average of two points.

3.9 Correlation

The Spearman's rank correlation coefficient is an approach for statistical dependence between two independent variables while not consider the parameters of a frequency distribution (nonparametric) (Zar, 1972; D. Liu et al., 2010; Gauthier, 2001). The rank correlation coefficient, r_s, can be expressed using a monotonic function as:

$$r_s = 1 - \frac{6 \sum a^2}{N(N^2 - 1)} \tag{3.1}$$

where r_s represents the Spearman's rank correlation coefficient, a denotes the ranked difference between the ith measurements for the two random variables, and N is the number of measurements in each of the two random variables in the correlation (Zar, 1972). Spearman's r_s measures the strength and course of the *monotonic* relationship between ranked variables. Hauke and Kossowski (2011), meanwhile, explain that Spearman's r_s can be considered as the correlation coefficient of Pearson that deals with ranks.

Pearson's product moment correlation coefficient, commonly Pearson's correlation coefficient or Pearson's r, is the most commonly used correlation statistic to measure the strength and course of the *linear* relationship between two continuous variables (Ahlgren et al., 2003). It is expressed using this function:

$$r = \frac{\sum(A - \bar{A})(B - \bar{B})}{\sqrt{\sum(A - \bar{A})^2 \sum(B - \bar{B})^2}} \tag{3.2}$$

where r represents the Pearson's r, A and B denote the two variables being observed, and $\bar{A}$ and $\bar{B}$ are the averages of the two variables respectively.

Both correlation analysis outputs a value between -1 to +1. If the result of a correlation is close to +1, then variable A has a positive relationship with variable B; vice versa, if the result is close to -1, then variable A has a negative relationship with variable B. If the result is approaching 0, then variable A has no relationship with variable B. Table 3.3 shows the correlation relationships with their corresponding ranges, where x is the exact correlation result.

Table 3.3: Results of Pearson and Spearman

Correlation Relationship	Value
very strong positive correlation	0.9 < x ≤ 1.0
strong positive correlation	0.7 < x ≤ 0.9
moderate positive correlation	0.5 < x ≤ 0.9
low positive correlation	0.3 < x ≤ 0.5
no correlation	-0.3 < x ≤ 0.3
low negative correlation	-0.5 < x ≤ -0.3
moderate negative correlation	-0.7 < x ≤ -0.5
strong negative correlation	-0.9 < x ≤ -0.7
very strong positive correlation	-1.0 < x ≤ -0.9

Although Spearman is considered a Pearson's r measure in terms of ranks, there are advantages that enable Spearman to be a more powerful measurement over Pearson (Gauthier, 2001). Spearman's rank correlation coefficient is not affected by the distribution of the values, unlike Pearson's r, which normal distribution is analyzed. Furthermore, instead of the raw data, it does not care for outliers because it deals with the ranks of the data. In addition, the data does not need to be in time-series, where the intervals are regular. In contrast, since Spearman disregards the outliers when it turns the data into ranks, there is a loss of information. Moreover, it is less powerful than the Pearson's r if the data is normally distributed.

Autocorrelation

Autocorrelation, sometimes called *serial correlation* or *lagged correlation*, is defined as the correlation of a time series to itself. The series of numbers arranged in time is correlated to the values of its own past and future. The purpose of autocorrelation is to find out if there is a pattern repeating in the data. The formula for autocorrelation is as follows:

$$r_k = \frac{\sum_{i=1}^{N-k}(X_i - \bar{X})(X_{i+k} - \bar{X})}{\sum_{i=1}^{N}(X_i - \bar{X})^2} \tag{3.3}$$

where k denotes the lags and N as the number of observations in the series.

3.10 Feature Engineering

Feature engineering is important for any system that uses machine learning. The performance of machine learning algorithms depends on how the data was presented to them. If they accept raw data directly, they would not give the intended results since machine learning algorithms do not have the capacity to extract insightful features from raw data automatically. This is where feature engineering comes in. It is using the domain knowledge of and from the raw data and create insightful features which would help machine learning algorithms to perform better.

A feature is a property that is common with all the independent units in the raw data. Choosing the right features for the model is of utmost necessity. To yield better results, models should be simple and be more flexible, which better features would produce.

The process of feature engineering is difficult and exhaustive. At first, a lot of features need to be decided and created regardless of its relevance. Afterwards, these features would be checked with the model to identify how and which features would help the model. Then, to avoid overfitting, feature selection should be used. This process would repeat until certain features are finalized and used with the model.

3.11 Fusion Techniques

To achieve the best performance, fusion techniques were made for designing systems which recognize patterns. By combining data, features, or decisions from a set of sensors, better inferences are made and accuracy is improved (Sohn &

Lee, 2003). Fusion techniques are applied in the field of military, where targets are automatically recognized (Bossé et al., 2006), in the field of image processing for medicine (Constantinidis et al., 2001), face recognition (Mangai et al., 2010), robotics (Jimenez et al., 1999), among others. There are three levels for fusion techniques: *data fusion, feature fusion,* and *decision fusion.* In this paper, only the feature fusion and decision fusion will be discussed. In addition, there are numerous techniques to solve fusion problems, but in this paper, only neural networks and weighted average will be discussed.

3.11.1 Fusion Levels

Feature Fusion

Features are the most essential information needed to accomplish any classification task. Although important, not all features contribute to the performance of a classifier. Features that do not match the rest of the dataset can have an effect to the pattern recognized by the classifier. Thus, they should be removed from the dataset (Mangai et al., 2010). The process of improving or obtaining new features is called feature fusion (Castanedo, 2013).

Dasarathy (1997) broke down the common hierarchy of fusion into five levels. In the said breakdown, he defined feature fusion as the Feature In- Feature Out Fusion (FEI-FEO). This fusion process accepts and outputs features. Feature fusion can be divided into three methods: feature ranking and selection, feature extraction, and combination (Mangai et al., 2010).

Mangai et al. (2010) described feature ranking and selection as the soul of feature fusion. It aims to find the optimal subset that can represent the entire dataset. In feature extraction, on the other hand, features are reduced and ranked for a better analysis Dasarathy (1997). Lastly, feature combination derives new features from at least two selected features.

Another fusion level that accepts features is the Feature In-Decision Out Fusion (FEI-DEO). Its difference with the FEI-FEO Fusion is that instead of having features as its output, it outputs a decision instead. Some researchers refer to this as either feature fusion because of its input or decision fusion because it outputs a decision. It all depends on the researcher's view (Dasarathy, 1997). It is commonly used by researchers for pattern recognition systems that generates a decision based on inputs from different sensors (Castanedo, 2013).

With the issues involving the validity of decision fusion for the affective field, D'Mello and Graesser (2010) used feature fusion for a multimodal affect detection system. The researchers combined features to generate a better distinction between common human experiences such as confusion, frustration, engagement, delight, boredom, and neutral. They concatenated features from three sensory channels to produce a multichannel feature set. As a result, they were able to generate four multichannel models: Facial-Dialogue (FD) model, Facial-Posture (FP) model, Dialogue-Posture (DP) model, and Facial-Dialogue-Posture (FDP) model. They used feature selection to filter out a number of features that have the highest F-ratio. These features were then combined with the set of features selected from the other channel.

Geetha and Radhakrishnan (2013) were able to get satisfactory classification results after fusing fingerprint and palmprint features for a multimodal biometric authentication system. It aims to identify the set of most important features that can be used for a better recognition accuracy.The researchers claimed that fusing the features extracted from 2D Gabon filter, stationary wavelet transform (SWT), and principal component analysis (PCA) by concatenation would result into a large feature set. This makes computing match scores more laborious. Hence, they used a wavelet-based fusion algorithm to fuse the extracted features. After using SWT to extract line information, the researchers used the mean-max fusion method to fuse the said features.

Decision Fusion

A decision is a conclusion stemmed from facts of a discerned events and activities (Castanedo, 2013). As the high-level fusion technique, the decision fusion is more complicated since it inputs decisions from multiple classifiers to obtain one accurate decision (Mangai et al., 2010). Dasarathy (1997) defines decision fusion as the Decision In - Decision Out (DEI-DEO) fusion in his five-step hierarchy of fusion based on input and outputs (see Figure 3.1). This combines multiple decisions from the input and outputs a more appropriate decision.

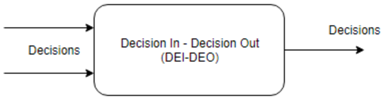


Figure 3.1: Dasarathy's Decision In - Decision Out (DEI-DEO)

Dasarathy (1997) also argues that while DEI-DEO fusion does not always trump data or feature fusion, the whole system does not fail if one of the sensors fail, unlike the other fusion techniques. He further explains that the DEI-DEO has less computational demands than the data or feature fusion, and that the bandwidth for communication is not important.

Castanedo (2013) states one of the most used methods for decision fusion. The first method is the *Bayesian Method.* This technique combines facts from probability theory rules, where the input and the outputs are probabilities in between [0, 1]. This method is derived from the Bayes rule:

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)}, \tag{3.4}$$

which gives us the probability of B given A. However, because the probabilities $P(A|B)$ and $P(A)$ are not always known, it may not be applicable to some cases. Furthermore, Hall and Llinas (2001) state that the Bayesian method has a complexity problem if there are more than one possible result for $P(B|A)$.

Other than Bayesian method, there are other techniques that can be applied to decision fusion, such as the Dempster-Shafer inference, abductive reasoning, semantic methods (Castanedo, 2013), logistic function, weighted decision methods (Sohn & Lee, 2003), projection pursuit, majority voting, fuzzy logic, and neural networks (Jimenez et al., 1999).

3.11.2 Fusion Algorithms

Neural Networks

One of the techniques that can solve fusion problems is through neural networks (Fincher & Mix, 1990). The advantage of having neural network as a technique for fusion over statistical approaches, such as Bayesian Method, is that it does not need to know certain probability errors beforehand. It can train itself with the current facts at hand and still get accurate results. In the following papers, they used neural networks to fuse multiple sensors by recognizing the patterns given the inputs via their supervised training algorithm such as perceptron and backpropagation.

In Fincher and Mix (1990), they used neural network-based data fusion for recognizing patterns between handwritten and computer-made numbers. The network is fed a 16 × 16 input character, and outputs the number recognized. They used two approaches in fusion: pre-detection and post-detection. In pre-detection, the input is fed into three algorithms: Quadrant Analysis, Outside-In

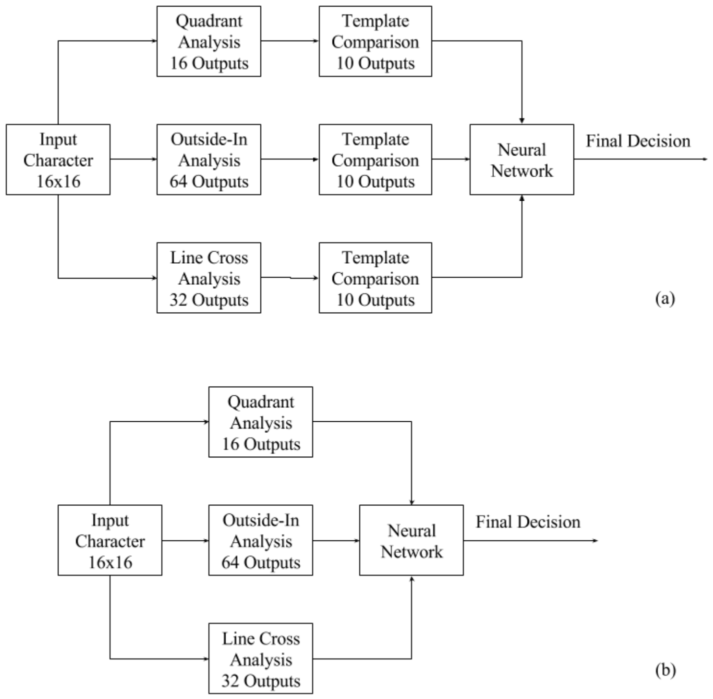

Quadrant Analysis 16 Outputs
Template Comparison 10 Outputs
Input Character 16x16
Outside-In Analysis 64 Outputs
Template Comparison 10 Outputs
Neural Network
Final Decision
Line Cross Analysis 32 Outputs
Template Comparison 10 Outputs
(a)
Quadrant Analysis 16 Outputs
Input Character 16x16
Outside-In Analysis 64 Outputs
Neural Network
Final Decision
Line Cross Analysis 32 Outputs
(b)

Figure 3.2: (a) Structure of the Fincher's Pre-Detection Approach, and (b) Structure of the Fincher's Post Detection Approach

Procedure, Line Cross Analysis. The outputs of these three algorithms are then compared to a template and creates decisions based from the algorithm outputs. These decisions are then fused through a neural network. In post-detection the outputs of the three algorithms are fed straight to the neural network to avoid loss of data. In this paper, pre-detection detected the correct pattern 93% of the time than post-detection which detected the pattern 86% of the time. The network bases its decision on the aspects of the input such as how close is one pixel to the other, and the importance of a value in a pixel. The network was trained through a supervised learning algorithm to merge and recognize the patterns based from the features of the input. The pre-detection follows the DEI-DEO fusion where the decision of the template comparisons are fused to generate the final decision, the pattern recognized from the inputs. On the other hand, the post-detection follows the FEI-DEO fusion where the input of the network is the features of the 16 x 16 input character and the output is the pattern recognized from the input. Figure 3.2 illustrates the pre-detection and post-detection approach.

Likewise, in Dai and Khorram (1999), they used neural networks for image processing to recognize changes between pictures of the same location of different times. The network is fed raw data from multiple sensors sequentially into 12 input layer units. Like (Fincher & Mix, 1990), the network bases its decision from the inputs given. Additionally, the network's hidden units play the role as the change extractor - one that can recognize and extract changes of the input and pass it to the next hidden layer. Finally, the network will output change map.

In (Jimenez et al., 1999), they compared different methods of decision fusion (DEI-DEO), including multilayer neural networks, in hyperspectral imaging. In their architecture, the inputs for their neural network are the decision outputs from different classifiers, whose inputs are vectors of a spectral band in an image. The network was trained through Perceptron supervised training algorithm. Merging the inputs via supervised learning, their output is a single decision obtained from the training of the neural network, identifying the object in the image.

Weighted Average

The Weighted Average is a signal-level fusion method that takes the weighted average of repeating information from the original inputs (King et al., 2017; Yan et al., 2011; Meurant & Meurant, 1992). King et al. (2017) claimed that using this method will ease the effect of unwanted data in the final estimation. This method can be used to estimate user activity details such as intensity, postures, and fundamental static for activity recognition systems.

Yan et al. (2011) proposed the use of the weighted average method on ships traffic flow data. They used data fusion to produce a higher estimation accuracy from multiple sensors. The researchers multiplied the number of ships with its corresponding weight then obtained the average of the produced weighted inputs. They then applied a distribution method to the original data to eliminate the unwanted data.

3.12 Neural Networks

3.12.1 Artificial Neural Network

Inspired from the nonlinear characteristics of Biological Brain System, the work performance of artificial neural networks (ANNs) can be compared to the workings of the human brain system (More et al., 2016). While handling uncertainty and non-linearity at the same time, ANNs provide "intelligent processing functions" in order to predict, learn, and memorize (Sommer et al., 2013). ANN can be applied to every situation where the input variables have a relationship with the output variables. The network for ANN consists of 3 layers in order, the input layer, hidden layer and output layer. Each layer consists of 1 or more units or neurons.

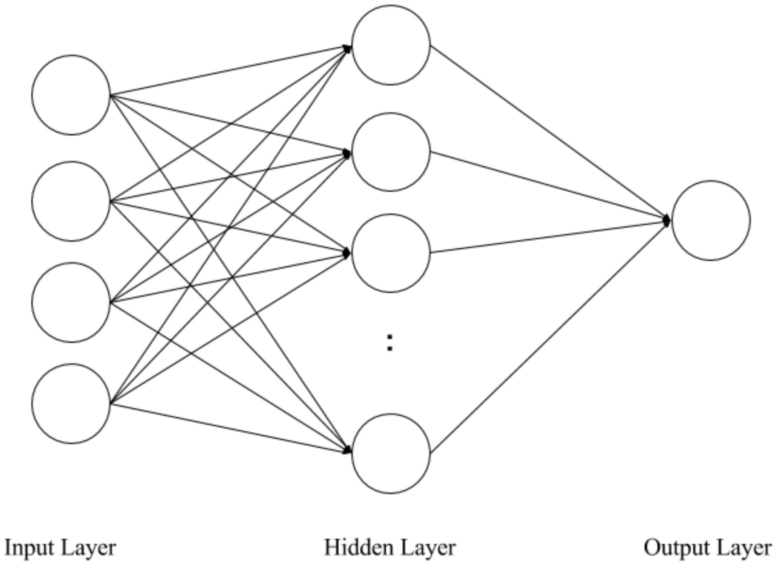


Figure 3.3: Network graph of a One-Hidden layer ANN

Figure 3.3 shows a the network graph of a One-Hidden layer ANN. Each set of inputs is modified by unique weights in unit connections, and biases in unit. Training in ANN involves adjusting all the weights and biases to have the correct output. A conventional ANN uses backpropagation for its training algorithm (Hamad et al., 2017). Backpropagation is a training algorithm which trains the neural network from the input layer to the output layer initializing the weights and biases, and then propagate back from the output layer to the input layer to adjust the weights and biases from the differences between the generated output to the expected output. In backpropagation, each unit that receives a value gets adjusted using an activation function. This training process is repeated a number of times until training error is close to optimal, or until the network reaches the maximum number of epochs. Once backpropagation is finished, the network is trained. The weights (w) are adjusted by many learning algorithms such as the Error Correction Learning where the connection weigths between unit i and unit j are adjusted in terms of the difference between the desired and the computed value. A multilayer error correction weight adjustment formula is defined as

$$w_{ij}^{new} = w_{ij}^{old} - \eta \frac{\partial E}{\partial w_{ij}^{old}} \tag{3.5}$$

where η is the learning rate - the percentage the network minimizes the error rate in each iteration and E is the square of errors of the desired output and actual output which can be defined as

$$E = \frac{1}{2} \sum_{j=1} (b_j - z_j)^2 \tag{3.6}$$

where b is the desired output value, and z is the actual output value.

3.12.2 Deep Belief Network

Koesdwiady et al. (2016) states that the training algorithm for ANN suffer from the problem of local minima, wherein the training algorithm may get stuck in the local minima during backpropagation. Deep Belief Networks (DBN) solve this problem by adding an extra step called pre-training which is done before backpropagation that can lead to an error rate not far from optimal. Backpropagation can be used then to slowly reduce the error rate from there. Additionally, other techniques and learning algorithms cannot extract and learn features without prior knowledge of specific domains. Deep Learning could learn features with less prior knowledge.

The training process of DBN consists of two phases, the pre-training and fine-tuning. In the pre-training process, the hidden layers of the DBN are trained greedy layer-wise. Pre-training generates the initially trained DBN with initialized weights and biases for each layer and unit. Fine-tuning further adjusts these weights and biases via backpropagation training algorithm using a labeled input data.

DBNs are made up of stacks of Restricted Boltzmann Machines (RBM). It is a building block for multi-layer learning models of Deep Belief Networks A DBN stack RBMs to learn features to features to arrive at a high-level representation. An RBM consists of 2 layers which are the visible layer with i visible units, and a hidden layer with j hidden units. The RBM is an unsupervised learning algorithm that can learn useful features of the data. It takes the input and translates them into a set of numbers that represents. Then, these numbers can be translated back to reconstruct the inputs. Through several forward and backward passes, the RBM will be trained. A trained RBM can reveal which features are the most important ones when detecting patterns (Zhang et al., 2017; Fischer & Igel, 2014). The hidden units of a trained RBM represent relevant features of observations. These features can also serve as input for another RBM. The network graph of an RBM is illustrated in 3.4.

RBM defines a probability distribution, via the energy function, between each

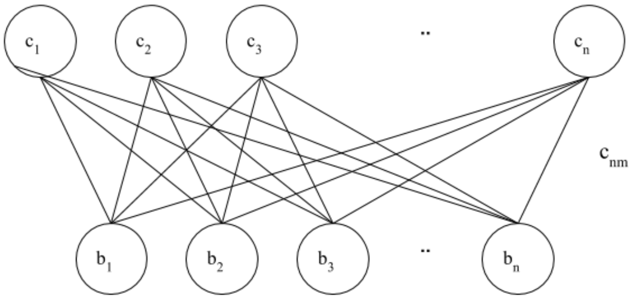


Figure 3.4: Network graph of RBM given n hidden units, and m visible units.

units' weights and biases. This distribution is defined as

$$p(v, h) = \frac{e^{-E(v,h)}}{Z} \tag{3.7}$$

where Z is the partition function, and $E(v, h)$ is the energy function defined as

$$E(v, h) = \sum_{i=1}^{V}\sum_{j=1}^{H} w_{ij}v_ih_j - \sum_{i=1}^{V} a_iv_i - \sum_{j=1}^{H} b_jh_j \tag{3.8}$$

where the v_i is the visible unit i, h_j is the hidden unit j, w_{ij} is the weight between v_i and h_j, a_i and b_j are their biases. V and H represent the number of visible and hidden units, respectively. The distribution of the visible units' weights and biases is defined as

$$p(v) = \sum_{H} p(v, h) = \sum_{H} \frac{e^{-E(v,h)}}{Z} \tag{3.9}$$

In RBMs, the units of the same layer are independent which do not have any connection with each other. In terms of probability, this means that the hidden variables are independent given the state of the visible variables and vice versa. Thus, the condition distributions of $p(h|v)$ and $p(v|h)$ are defined as

$$p(h_j = 1|v) = \sigma\left(\sum_{i=1}^{V} w_{ij}v_i + b_j\right) \tag{3.10}$$

$$p(v_i = 1|h) = \sigma\left(\sum_{j=1}^{H} w_{ij}h_j + a_i\right) \tag{3.11}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$, the sigmoid function of x, the feature, element-wise.

In DBN's pre-training, each RBM is trained individually, greedy layer-wise. The weights of each units' connections, and each layer's biases, are computed within this phase and saved for fine-tuning. This learning is repeated, from RBM to another, until all hidden layers are trained. This process can be expressed as follows

$$p(h_j^{(l-1)} = 1|v) = \sigma(\sum_{i=1}^{V} w_{ij}^{(l)} v_i^{(l)} + b_j^{(l-1)}) \tag{3.12}$$

where l corresponds to the current layer. The formulas and equations for RBM are based from the papers of Koesdwiady et al. (2016); Fischer and Igel (2014); Zhang et al. (2017).

3.13 ParseHub

ParseHub is a free web scraping tool that collects from any JavaScript or AJAX site. This web scraping tool can also export the collected data in CSV and JSON files. The tool also has the ability to cloud host which enables the data scraped to be saved through the cloud, and schedule a run to a future time. ParseHub can scrape from hundreds of pages depending on the pricing. The Free ParseHub may only collect 200 pages worth of data in under 40 minutes. It also can run 200 pages per run. Lastly, ParseHub retains data collected for 14 days.

3.14 Performance Indices

3.14.1 RMSE and MAE

To evaluate a model's performance, many studies have employed using two of the most used performance indexes: the root mean squared error (RMSE) and the mean absolute error (MAE) (Chai & Daxler, 2014). The RMSE is used as a standard statistical system of measurement to calculate a model's performance, especially in air quality, meteorology, and climate research studies. It is defined using this function:

$$r = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2} \tag{3.13}$$

where n is the number of samples, x_i and y_i are the errors being observed at a certain i. On the other hand, MAE is a measure of the absolute difference between the errors. It is defined using this function:

$$r = \frac{1}{n}\sum_{i=1}^{n}|x_i - y_i| \tag{3.14}$$

3.14.2 Sensitivity Analysis

Sensitivity analysis is a study of how much the input variables clearly affect the outcome of a model (Saltelli et al., 2000). It is also called the simulation analysis, wherein the results are predicted based on a range of input variables. Calculating the sensitivity that is dependent on time, we calculate the sensitivity relative to the initial parameters, inputs, or conditions in the model.

The "time-dependent" sensitivities of output Y relative to every input variable are "time-dependent" derivatives is given by the equation:

$$\left|\frac{\partial Y}{\partial X_i}\right|_{X^0} \tag{3.15}$$

where, Y is the output and X_i is the input factor. X^0 indicates that the derivative is taken at some fixed point in the space of the input. This shows that per input of the model, the ratio of its derivative with respect to its output is computed to get the sensitivity.

Several researchers have been applying sensitivity analysis to their models, particularly to the models which uses neural networks. In Hunter et al. (2000), they applied sensitivity analysis of neural network inputs to their trauma survival prediction model to analyze the Trauma and Injury Severity Score (TRISS) variables used in their model. Their experiments on the variable's sensitivity show that age is the most prominent variable in predicting survival. The researchers defined three approaches to sensitivity analysis. First is to introduce noise to every input variable and to observe its effects to the results. Second is to examine the derivatives of the weights of the input variables. Last is the proposed form of analysis based from the missing value problem approach.

In Refenes et al. (1994), they performed sensitivity analysis to evaluate their model on stock performance using neural networks. They used scatter plots to visually analyze the change in their network output (the likelihood of a sample being a target) with respect to the network input (stock factors). Based on their result, they were able to explain the predictive behavior of their neural network output by using sensitivity analysis. Furthermore, they were able to conclude that compared to regression models, their model can model the situation more convincingly.

Chapter 4

Research Design

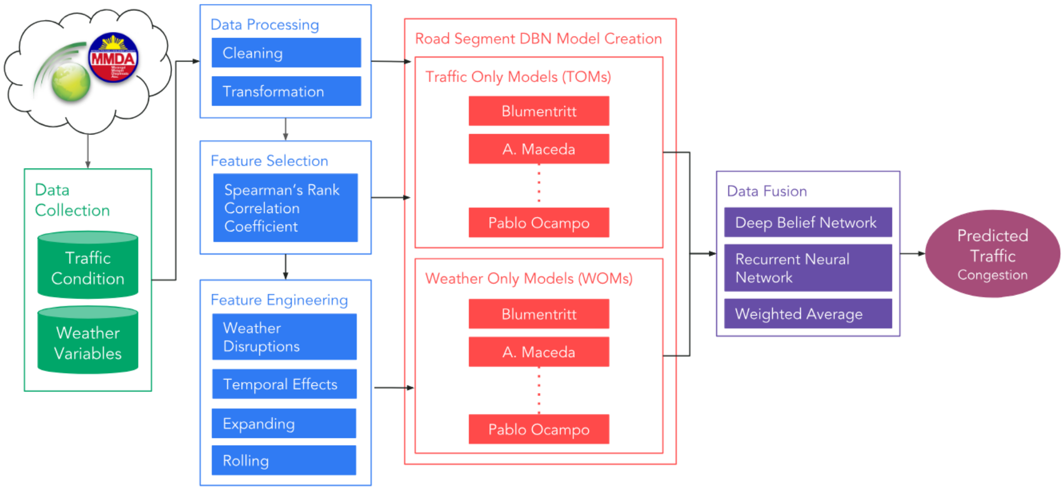


Figure 4.1: Research Framework

Figure 4.1 illustrates the whole framework for the research. First, this research collected historical traffic condition from MMDA and historical weather variables from WWO. These raw data was cleaned and re-sampled to match the time intervals of the data using linear interpolation and was transformed to their normalized values. Both datasets were cross-correlated with each other to analyze and determine significant weather variables. Aside from that, the effects of extreme weather disruptions to traffic were explored. Furthermore, that may neglect the effect of weather on traffic were also explored. From these analyses, features were extracted that would better represent the data for our problem through *feature engineering* using rolling and expanding.

The final variables and engineered features were fed into the Traffic-Only Models (TOM) and Weather-Only Models (WOM), which were developed in DBN, for 14 roads in Manila. Following the DEI-DEO fusion technique, the predicted traffic congestion for TOM and WOM were fused in a fusion model, in which three methods of fusion was compared. The model was evaluated using RMSE, MAE, and sensitivity analysis. To achieve a high accuracy, the model's parameters were calibrated based from previous experiments.

4.1 Data Collection

There were two datasets collected: traffic and weather. The traffic dataset was obtained from Metro Manila Development Authority (MMDA) traffic monitoring system. The weather dataset was collected from World Weather Online (WWO). Both datasets were collected from May 2015 to October 2015, highlighting the wet season in the Philippines.

4.1.1 Traffic Dataset

There were two limitations found from the traffic dataset. First is that the traffic conditions were only represented as *light* (L), *moderately light* (ML), *moderate* (M), *moderately heavy* (MH), and *heavy* (H). Since this dataset only had five values to classify the traffic condition in a particular road segment, this might cause underfitting in our model since it was used as input for a regression problem. Moreover, it could also contribute to poor correlation as these five values were correlated with continuous weather variable values. To have the traffic conditions be continuous values, they were converted to their equivalent estimated traffic speed provided by the MMDA (see Table 4.1). These speeds are further converted to congestion levels, represented as their reciprocal so that the data would be easier to interpret such that higher value means higher congestion level.

Table 4.1: Traffic speed equivalent of traffic conditions

Traffic Condition	**Equivalent Speed (kph)**
Light (L)	36 - 60
Moderately Light (ML)	31 - 35
Moderate (M)	16 - 30
Moderately Heavy (MH)	11 - 15
Heavy (H)	0 - 10

Another limitation was that the traffic dataset contains missing records. Out of 4,749,417 records, there were 522 rows wherein the traffic condition of that road segment in that particular time interval was not recorded, as indicated by *none* (N) condition. Furthermore, having only 4,749,417 records from a sample from 2015, having a 15-minute interval for 142 road segment, meant that 226,263 records or 16 days of data were missing as well. This implied that inconsistencies may occur as the missing data would need to be filled to make the data continuous. To do this, those missing values were replaced through linear interpolation.

Apart from the limitations, there were other factors considered in this study. Since this study is only concerned road segments from Manila, only the road segments under it were used. As a result, 14 out of 142 road segments were only considered: 7 from Roxas Boulevard and 7 from España.

4.1.2 Weather Data

The collected weather dataset from WWO featured a complete hourly reading of the weather variables for Manila. One downside of this dataset, however, was that the weather data generalizes the weather for the whole city. This implied inconsistencies when correlating the traffic condition to the weather variables as the weather at one road is not the same as the weather of another, despite being in the same city.

Additionally, the hourly weather dataset needed to be matched with the 15-minute interval of the traffic dataset. This was done by resampling and linear interpolating the dataset to have a 15-minute time interval.

4.2 Exploratory Data Analysis

To be able to extract the relationship between these two independent datasets, we need to understand the underlying pattern of each other and its effect on one another. In this study, we approached this in three steps. First, patterns of traffic and weather are analyzed as independent entities. From this, patterns are compared to identify the relationship between these two entities. Once these patterns are bridged, instances of weather disruption to traffic are derived by associating the weather variables that may be a contributing factor to traffic.

4.2.1 Weather

Weather is theoretically unpredictable. Nevertheless, it is known to have a yearly pattern such that the dry season is expected to occur around December to May, and the wet season to occur around June to November (*Climate of the Philippines*, n.d.). Therefore, in spite of its unpredictability, there is still an underlying pattern within it. In these analyses, we examine each weather variables in two ways: identifying its seasonality in the short-term (daily and weekly), and identifying its daily trend.

Seasonality

Seasonality refers to a predictable pattern of a time series data that regularly repeats after a number of intervals. To identify the short-term seasonality of weather variables, we analyzed the similarity between an observation and time lags between them through autocorrelation. Lags refer to the delay of one observation on another. For instance, the delay between the temperature observations at June 18 and June 19 is indicated by a one-day lag.

Assessing the seasonality of weather variables is useful in our study as it allows us to identify the span of the trend of a specific weather variable. Observably, one weather variables that do have a daily pattern is temperature. Its pattern starts by rising in the morning, peaking in the afternoon, then starts falling in the evening. There are also weather variables, on the other hand, that does not have an observable pattern. One example of this is precipitation. Unlike temperature that has an observable pattern per day, the occurrence of precipitation is situational.

Trend of Weather Variables

Once we are able to identify the short-term seasonality of individual weather variables, we zoom in further in given observations, based on the identified seasonality time spans, and analyze how similar are these in terms of trend. For instance, we will analyze how similar are temperature on one observation and its previous day, and how different are precipitation on one observation and its previous day.

4.2.2 Traffic Analysis

On an average day, traffic could be linked with the people's daily transportation demand. One example of these is the daily commute of people, attributed to their organizational duties (i.e. 9 to 5 jobs) and academical duties. Built upon these expected demands, the daily seasonality of traffic has been established, following the concept of peak hour. Peak hour refers to the busiest hour where traffic is expected to rapidly rise. In the case of Metro Manila, peak hours are expected to occur from 7 AM to 10 AM, when people leave their home to go to their respective organizations, and 4 PM to 7 PM, when they depart their organizations and return home (Regidor, 2013).

However, knowing that there are other contributing factors to traffic, its pattern is inconsistent despite its people's daily commute schedule remains constant as the years go by. Given this problem, it is significant to identify the more accurate estimate of the seasonality of traffic, considering that these contributing factors are already embedded in the day-to-day traffic. These could be done by analyzing a road segment, as an individual point-to-point segment, and analyzing a road segment, as a part of the whole road, taking into account its connected roads.

Immediate Past Traffic (Rolling and Expanding Features)

Given that traffic shows seasonality and knowing that there are possibilities of sudden disruptions in traffic, we explore the effect of immediate past traffic to the current traffic condition. In order to get a bigger look on the the effect of the immediate past traffic conditions to the current traffic condition, rolling and expanding traffic features were generated for window sizes *4, 8, 24, 48*, and *96*, with each window representing a 15-minute time interval. Thereby, the generated window sizes translates to *1, 2, 6, 12*, and *24*-hours time interval. The statistical features such as the mean, minimum and maximum for both rolling and expanding windows were generated to describe the immediate past traffic conditions given a specific window size.

In order to analyze the relationship between traffic trends, the autocorrelation of traffic was generated, and the cross correlation between the original traffic and the engineered features of rolling and expanding was analyzed. From there, the rolling and expanding traffic features were used as additional features for TOM.

4.3 Model Implementation

4.3.1 Prediction Models

According to the framework in 4.1, two models were implemented using Deep Belief Network (DBN): the Traffic-Only Model (TOM) and the Weather-Only Model (WOM). TOM considers historical traffic condition intensity to predict traffic, while WOM only considers historical weather variables to predict traffic. In this study, DBN models were implemented with the open source machine learning framework in python, *tensorflow*. The architecture of the model for TOM and WOM are different, and suits the dataset used. For TOM, there are two hidden layers with 14, and 390 units respectively. TOM was trained with 15 epochs during the pre-training phase, and 150 epochs during the fine-tuning phase. For the WOM, there are 3 hidden layers with 7, 70, and 100 units respectively. WOM was trained with 11 epochs during the pre-training phase, and 150 epochs. The learning rate for both DBN and RBM for both models is 0.01. The activation function for both models is the *relu* function.

Deep Belief Networks (DBN) are made up of stacks of Restricted Boltzmann Machines. The training for DBN consists of two phases: pre-training and fine-tuning. In pre-training, each RBM is trained individually, and weights and biases of layers are fixed. In fine-tuning, the weights and biases of the whole network are updated via back propagation using labeled input data.

Figure 4.2 shows the process of how TOM and WOM using DBN was trained for prediction. The training process consists of three phases. In the first phase, the stacks of RBM (sRBM) within the network is trained individually. At the end of this phase, the weights and biases of the whole DBN are initialized. In the second phase, the DBN fine tunes the initialized weights and biases using backpropagation algorithm. The network after stage 2 is the trained and enhanced DBN. In the third phase, the network predicts the future traffic condition using a testing dataset.

For the pre-training phase, the sRBM performs a number of forward and backward passes until the reconstructed data is close to the original input. The process of backward and forward passes is performed using the equations 3.10 and 3.11 to compute for the conditional probability of the hidden unit h_j and visible unit v_i, respectively.

For the fine-tuning phase, the loss function and activation function is defined. The loss function for the softmax layer to improve the learning rate uses the cross-entropy equation between the visible and hidden units. The cross-entropy

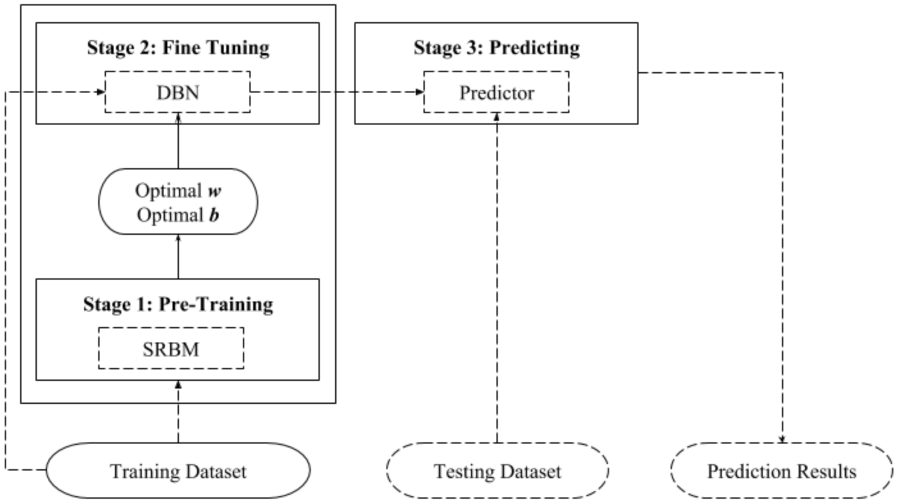


Figure 4.2: Structure of DBN Training Process

function is defined as

$$C = -\frac{1}{n}\sum_{x}[ylna + (1-y)ln(1-a)] \tag{4.1}$$

where n is the size of the training data, x is the training input, y is the desired output, and a is the unit's output. For the activation function in the fine-tuning phase, the rectifier function was used for its nonlinearity activation capability. The rectifier function is defined as

$$f(x) = x^{+} = max(0, x) \tag{4.2}$$

4.3.2 Data Fusion Model

This study's framework will evaluate two fusion approaches: Feature In-Decision Out (FEI-DEO) and Decision In-Decision-Out (DEI-DEO). In the first approach, traffic and weather features are fused in one dataset first before being used to predict traffic. While the latter approach predicts traffic through TOM and WOM, and fuses the prediction of both models into one final prediction. DBN is used in the FEI-DEO approach, while three algorithms are used in the DEI-DEO approach which are Weighted Average (WA), Recurrent Neural Network (RNN) and DBN.

Weighted Average

The weighted average data fusion method used the predicted traffic from TOM and WOM. Both predictions was multiplied with its corresponding weight. Each weight was assigned based on the importance of the variable considered while keeping in mind the sum of the weights should be equal to 1. Then, the mean of the weighted predictions of both TOM and WOM will be calculated. The result will be the fused final traffic prediction.

Neural Network

The fusion model was also developed in two neural networks: Recurrent Neural Network (RNN) and Deep Belief Network (DBN) in evaluating DEI-DEO fusion approach. Only DBN was implemented in evaluating FEI-DEO approach.

In FEI-DEO, the traffic congestion intensity and weather variables were fused into one dataset before feeding it into the prediction model. The network will be trained to predict the traffic condition based from both traffic and weather in one model for the current time period t. The training of the network will continuously adjust weights and biases from comparing the generated output with the actual traffic condition.

The DBN for FEI-DEO fusion have 1 input layer, 3 hidden layers, and 1 output layer. There are 5 epochs for the pre-training phase, and 150 epochs for the fine-tuning phase. The learning rate for both DBN and RBM is 0.01.

In DEI-DEO, the traffic congestion intensity was predicted first with two prediction models, TOM and WOM. The prediction by these two prediction models was used into another model that will fuse the two predictions into one improved final prediction. The network will be trained to predict the traffic congestion intensity for the current time period t through backpropagation by comparing the generated traffic condition prediction to the expected prediction, and adjusting the weights and biases of units and layers to fuse the two decisions to arrive to a prediction close to the expected. The input layer will have the predicted traffic condition of TOM and WOM.

The RNN for DEI-DEO fusion have only have 1 input layer, 1 hidden layer, and 1 output layer. The number of units for the hidden layer of the RNN will be determined through trial and testing starting from 5 to 100 units.

The DBN for DEI-DEO fusion will only have 1 input layer, 3 hidden layers, and 1 output layer. The DBN for the DEI-DEO fusion will be similar with the TOM and WOM in terms of it's hidden layer structure, and epochs. However, instead of historical traffic and weather data as its input, predictions made by TOM and WOM is used and fused into one final prediction.

4.4 Training

The collected dataset for traffic and weather was divided into two subsets; wet and dry season. The wet season dataset consists of data from the months of May to October 2015, while the dry season dataset consists of data from the months of November to April 2015. The models was tested on these datasets to evaluate the inclusion of weather variables.

Datasets was split into training and testing datasets. The Training dataset was used during the pre-training and fine-tuning of the model. In the fine-tuning phase of the training of the model, the training dataset was labeled with the expected output so as to verify and adjust the weights and biases back from the pre-training. The labels consists of the expected traffic condition given the input data. All data fed into the network are already be normalized.

The training dataset for both prediction models Traffic Only Model and Weather Only Model consists of data from the months of May to August was used for evaluating wet season, and months of January to April was used for evaluating dry season. In turn, the testing dataset made use of the remaining months of the season: September to October for the wet season, and November to December for the dry season.

The training dataset for the Traffic-Only Model consists of the following information in a 15-minute time interval: Temporal Information of the respective time period (i.e. month, hour, minute, day, and day of week), Traffic condition intensity a day before the respective traffic, Flags on working day and peak hour of the respective time period , Rolling and expanding window features (which consist of the minimum, maximum, and mean for each window) for the immediate past traffic of the respective traffic, Mean of the traffic 2 weeks before the respective traffic; In further experiments, the inclusion of the information regarding working days, peak hours, and immediate past traffic was compared and evaluated. The label for the training dataset consists of actal the 15-minute time interval of the traffic condition intensity of the respective time of the bound of the road segment to be predicted. For visualization purposes, the input including the labels is illustrated in Figure **??**

Caption: A summary on the contents of the Traffic Only Model Training and Testing Dataset

The training dataset for the Weather-Only Model consists of the following information in a 15-minute time interval: Temporal Information of the respective time period (i.e. month, hour, minute, day, and day of week), and; Current weather variables (e.g. temperature, precipitation, wind speed, etc.) of the respective time period In further experiments, different combinations of weather variables is compared and evaluated. The label for the training dataset consists of the actual 15-minute time interval of the traffic condition intensity of the respective time of the bound of the road segment to be predicted.

For the feature-level fusion model, the training dataset consists of merged training dataset used in the Traffic-Only and Weather-Only Models. The months for the training and testing datasets for this fusion model is the same as Traffic-Only and Weather-Only Model.

For the decision-level fusion model, the training dataset consists of the predicted traffic for the month of September and November generated by the Traffic-Only and Weather-Only model. The testing dataset consists of the predicted traffic for the month of October and December. The label for the training dataset will consist of the 15-minute time interval of the actual traffic condition intensity of the respective time of the bound of the road segment to be predicted.

4.5 Evaluation

The performance of the all models and algorithms (DBN, RNN, and WA) was evaluated using RMSE and MAPE. The performance of the fusion model fusing in the feature level (FEI-DEO) and in the decision level (DEI-DEO) was first evaluated. In evaluating DEI-DEO, the performance of TOM and WOM models were first evaluated before the fusion model/s. To evaluate the performance of TOM and WOM, different feature combinations are were. The performance of the model with these different feature combinations are compared with each other to see which input variables are needed in predicting the traffic condition intensity in current time period t. Then, three fusion techniques, namely weighted average, RNN and DBN, were compared and evaluated to see the accuracy of the final prediction. Additionally, the performance of the TOM and the final prediction of the DEI-DEO fusion model was evaluated to see relevance of the inclusion of weather variables as a factor in predicting traffic.

The model's sensitivity was also explored to analyze the relevance of input variables. Sensitivity analysis is also performed to achieve higher accuracy through calibration of hyper parameters. Specifically, the sensitivity for the models TOM, WOM and DEI-DEO fusion center in DBN will be evaluated. In evaluating the sensitivity for TOM and WOM, different combinations of input variables will be evaluated to see how much the inclusion and removal of a variable affects the final prediction. The parameters of the network was also experimented with, specifically the number of hidden layers and its units, the number of epochs for pre-training and fine-tuning and the batch size was tinkered.

Chapter 5

Results and Discussion

This chapter discusses the analysis of the traffic and weather data. Moreover, this chapter will also discuss about the experiments performed and its results to evaluate the analysis of traffic and weather.

5.1 Traffic and Weather Analysis

5.1.1 Weather Analysis

Seasonality

In spite of having a daily pattern, however, there are weather variables whose pattern would not last for every day of the year or even a week. One of these weather variables is precipitation. In the autocorrelation of precipitation (see Figure 5.1), it could be observed that despite having a daily seasonality, the pattern of the previous day cannot be barely used to predict the amount of precipitation to be expected today, as indicated by the weak correlation as it approaches its daily seasonality. Comparing it to the autocorrelation of temperature (see Figure 5.2), the pattern of the previous day or even until a week ago can be used to predict the incoming temperature of the day, as indicated by the consistently strong correlation throughout the week.

With these weather variables having a shorter term pattern, these could have situational effects to related or dependent variables whose usual pattern may be disrupted. Thus, we classify these variables as *disruptive variables.* These include

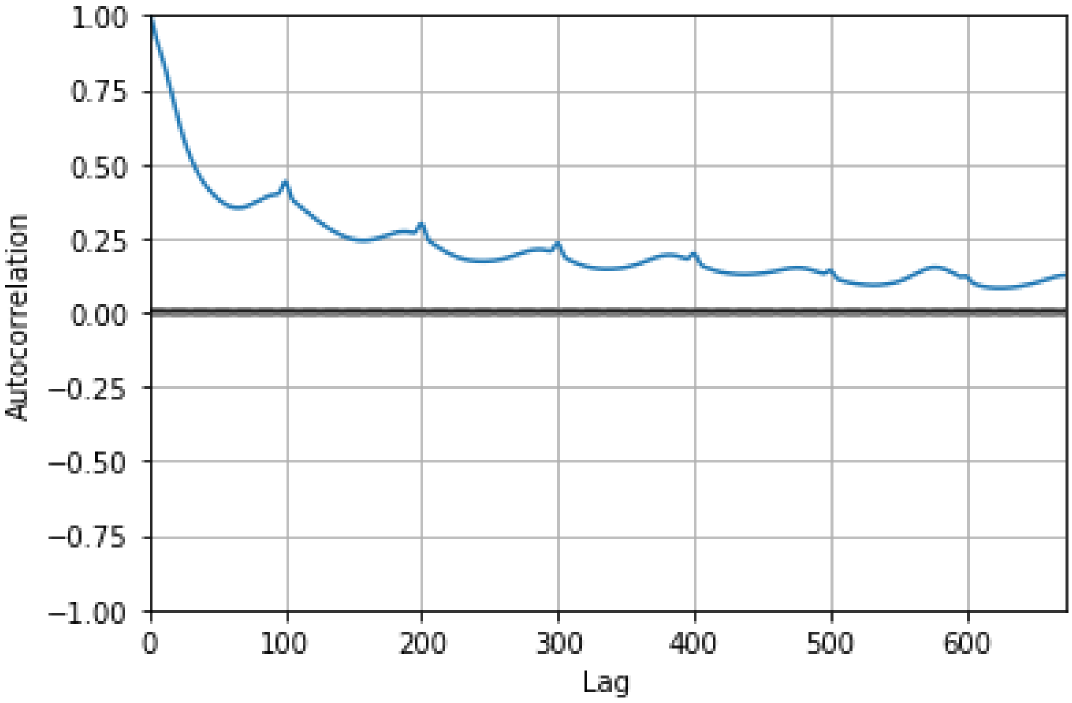


Figure 5.1: Autocorrelation of precipitation showing a loose relationship with its daily seasonality (1 interval = 15 minutes)

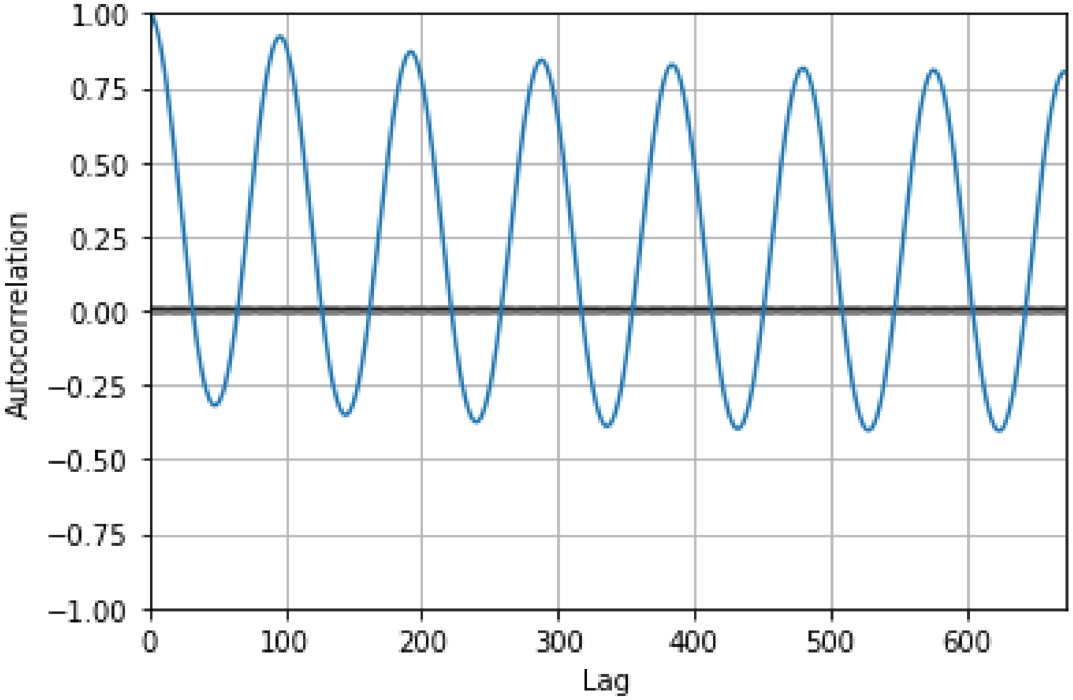


Figure 5.2: Autocorrelation of temperature showing a strong relationship with its daily seasonality (1 interval = 15 minutes)

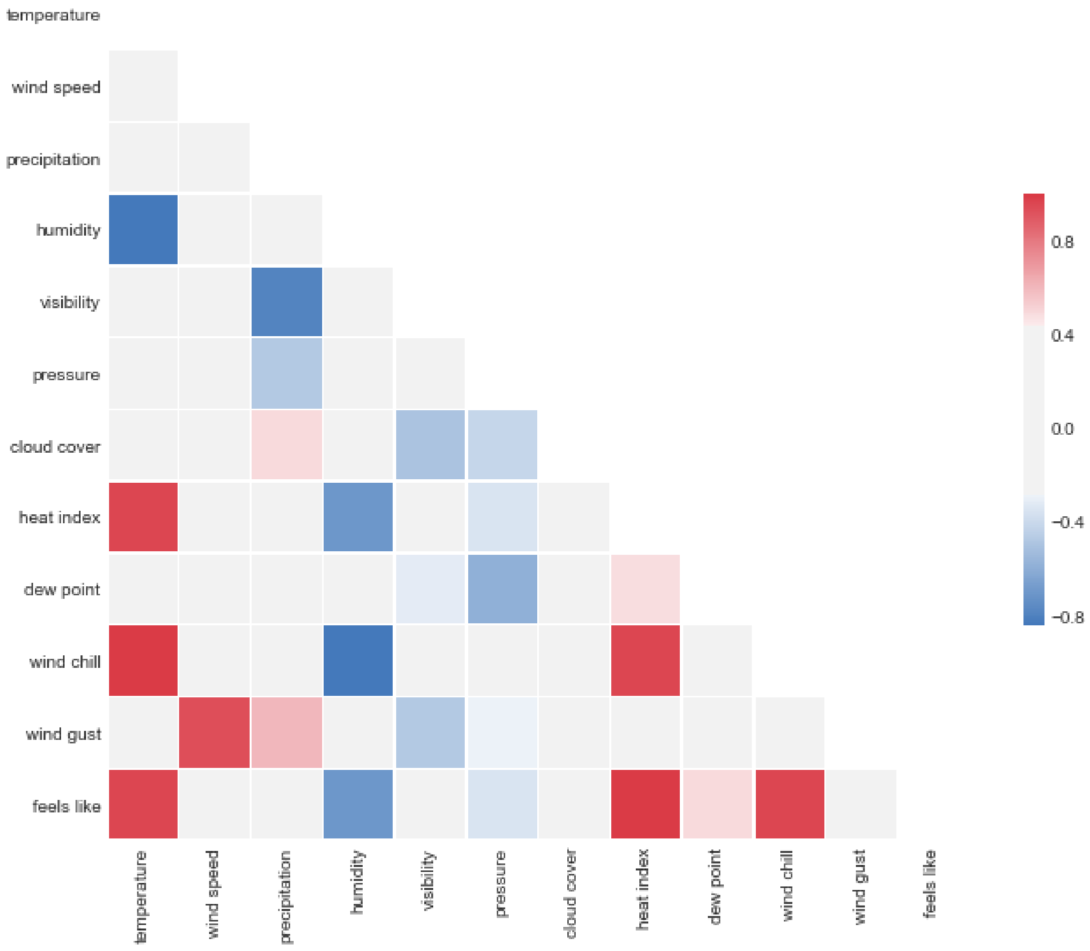


Figure 5.3: Correlation heatmap of weather variables

wind speed, wind gust, precipitation, and visibility (see Figure **??**). Looking at the multicollinearity of these variables (see Figure 5.3), it could be observed most disruptive variables have a strong correlation with precipitation. Particularly, both visibility and wind gust are strongly correlated with precipitation, having a correlation value of -0.787 and 0.600 respectively; and wind gust is strongly correlated with wind speed, with a correlation value of 0.934.

Trend of Disruptive Variables

Although these disruptive variables have a shorter term pattern, some still follow certain trends despite being influenced by other disruptive variables. One of which is wind gust (see Figure 5.4). Similar to non-disruptive variables, wind gust

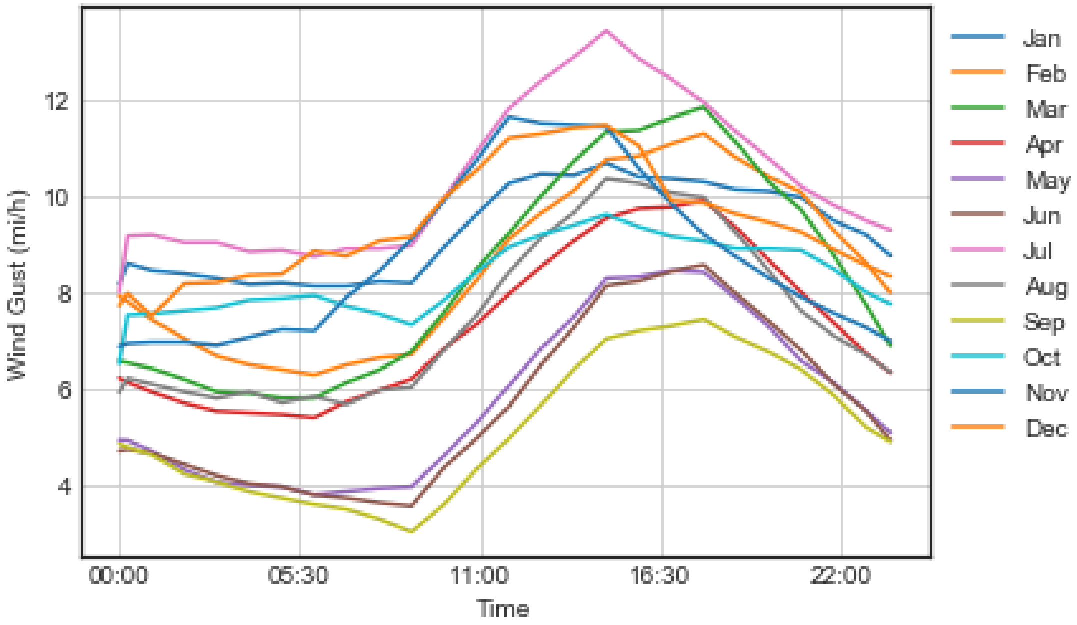


Figure 5.4: Daily average of wind gust per month revealing its normal trend despite having a loose daily seasonality

follows a long-term pattern yet is heavily impacted by other disruptive variables, specifically precipitation. It follows the trend of gradually falling until noontime, rising and peaking during the afternoon, and falling at a similar rate during the evening.

In instances where precipitation is abundant, however, there is an immediate effect to its pattern. Figure 5.5 and 5.6 shows a comparison of the impact of precipitation to temperature and wind gust in one of the days under Typhoon Egay. It could be observed that the precipitation did not significantly impact the pattern of the temperature. Nevertheless, we could see how it gradually decreased the temperature from its usual pattern in that specific month. Comparing it with wind gust, it could be observed that the effects of precipitation already took an effect on the pattern of wind gust as it starts to build up. Wind speed, being strongly correlated with wind gust, follows this trend as well (see Figure 5.8).

In a similar way, precipitation also affects other disruptive variables, mainly visibility. Contrary to wind gust and wind speed, visibility does not have a long-term pattern. Rather, it has a strong negative correlation with precipitation, hence its short-term trend is dependent on precipitation instances (see Figure 5.9).

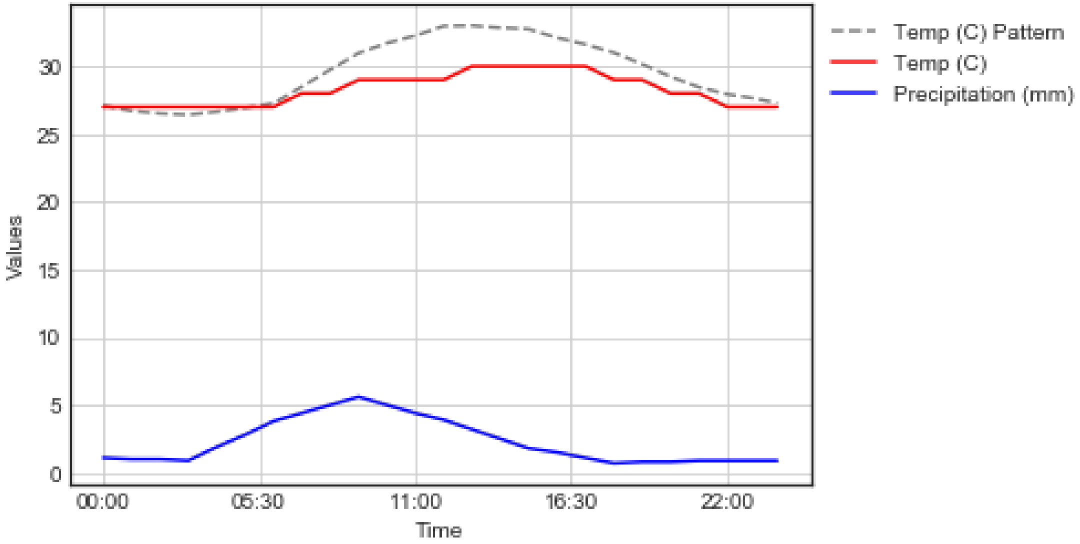


Figure 5.5: Comparison of the effects of precipitation to the normal pattern of temperature

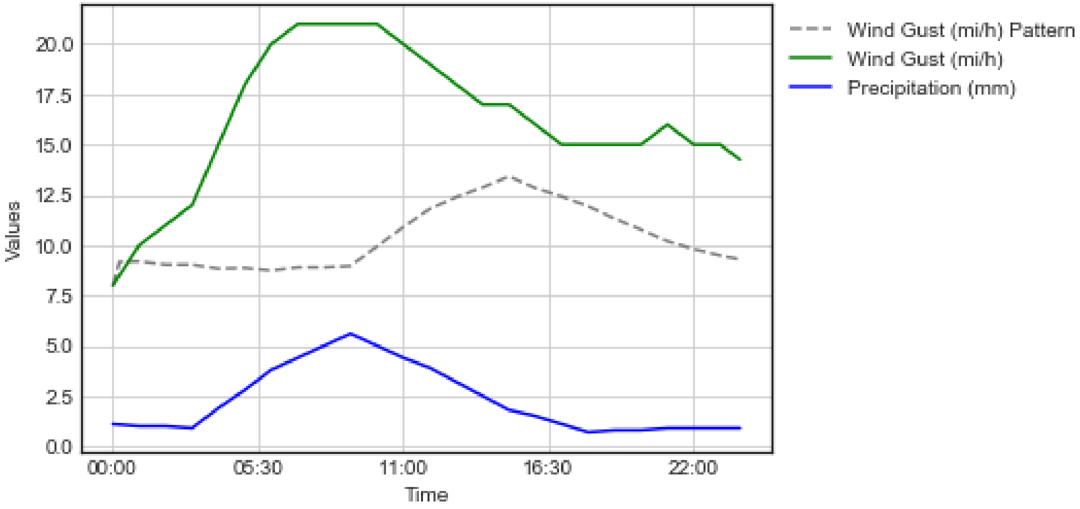


Figure 5.6: Comparison of the effects of precipitation to the normal pattern of wind gust

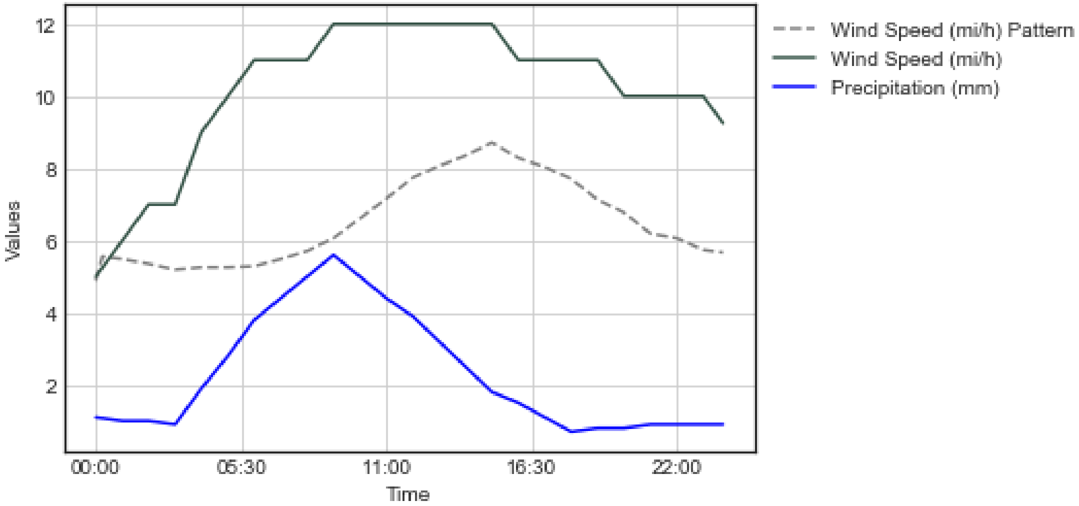


Figure 5.7: Comparison of the effects of precipitation to the normal pattern of wind speed

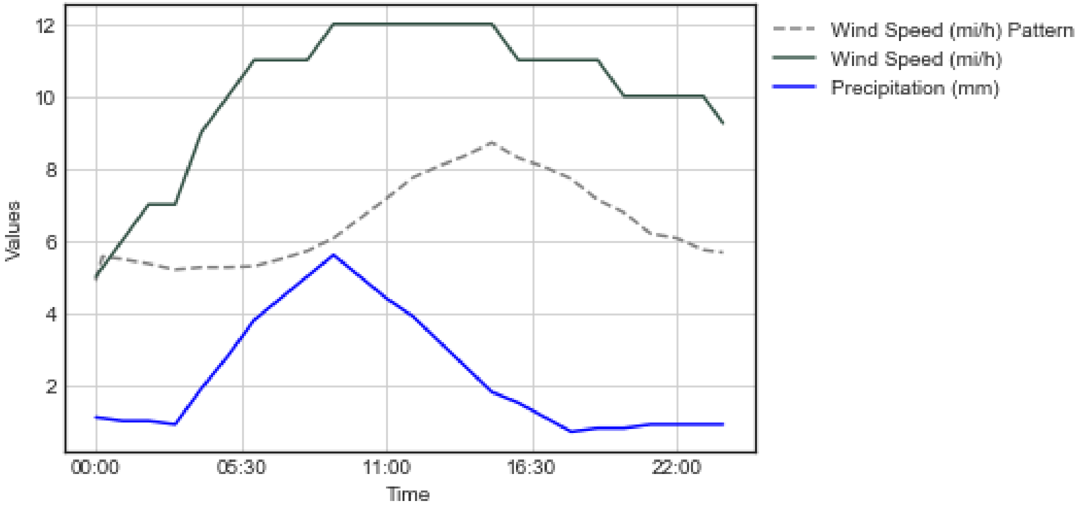


Figure 5.8: Comparison of the effects of precipitation to the normal pattern of wind speed

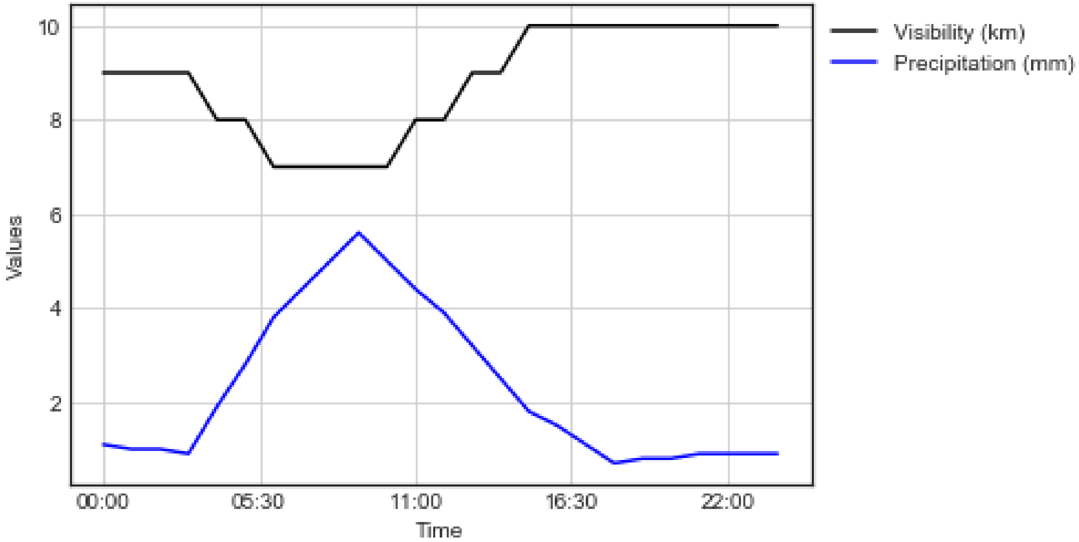


Figure 5.9: Negative relationship between visibility and precipitation

5.1.2 Traffic Analysis

Seasonality

On an average day, traffic could be linked with the people's daily transportation demand. One example of these is the daily commute of people, attributed to their organizational duties (i.e. 9 to 5 jobs) and academical duties. Built upon these expected demands, the daily seasonality of traffic has been established, following the concept of peak hour. Peak hour refers to the busiest hour where traffic is expected to rapidly rise. In the case of Metro Manila, peak hours are expected to occur from 7 AM to 10 AM, when people leave their home to go to their respective organizations, and 4 PM to 7 PM, when they depart their organizations and return home (Regidor, 2013).

However, knowing that there are other contributing factors to traffic, its pattern is not consistent despite its daily seasonality. Analyzing its autocorrelation (see Figure 5.10), it could be observed that in spite of its daily pattern, its succeeding days are loosely correlated with the present one. Interestingly, though, the traffic pattern of the previous day could also be used as much as the succeeding days or even the week after. Furthermore, compared with the previous days, it could be observed the present traffic is more correlated with its 7-days-ago pattern or its previous week (e.g. relate the present Monday with last week's Monday).

Viewing its pattern in a longer term, nevertheless, its weekly pattern becomes

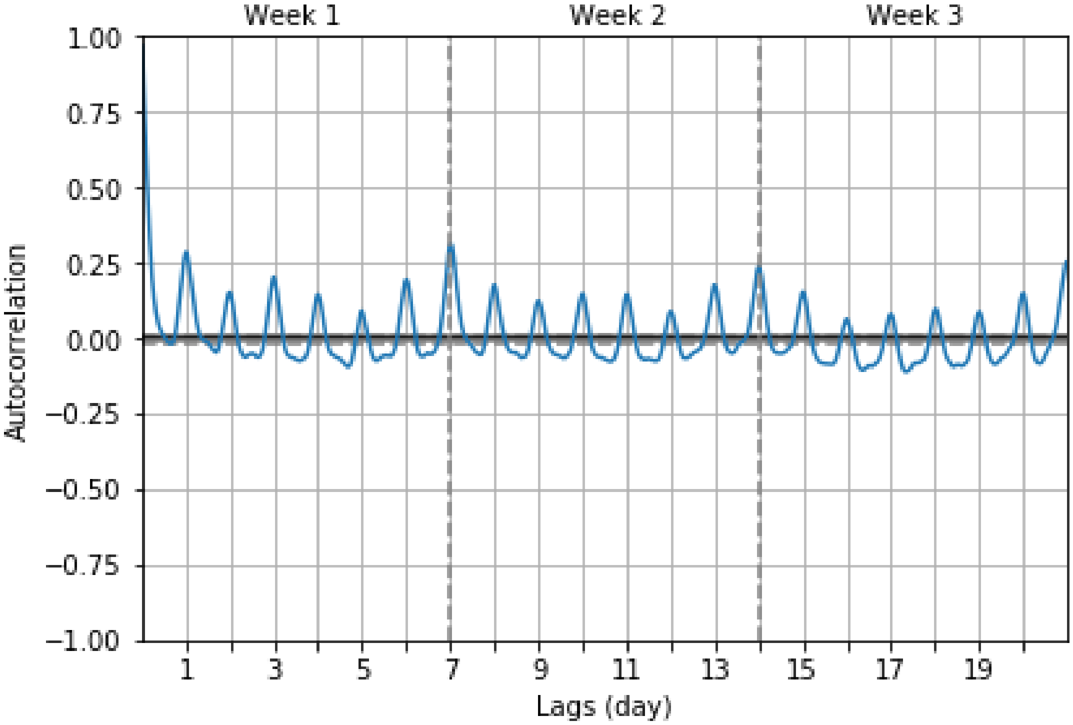


Figure 5.10: Autocorrelation of traffic revealing its daily and weekly seasonality

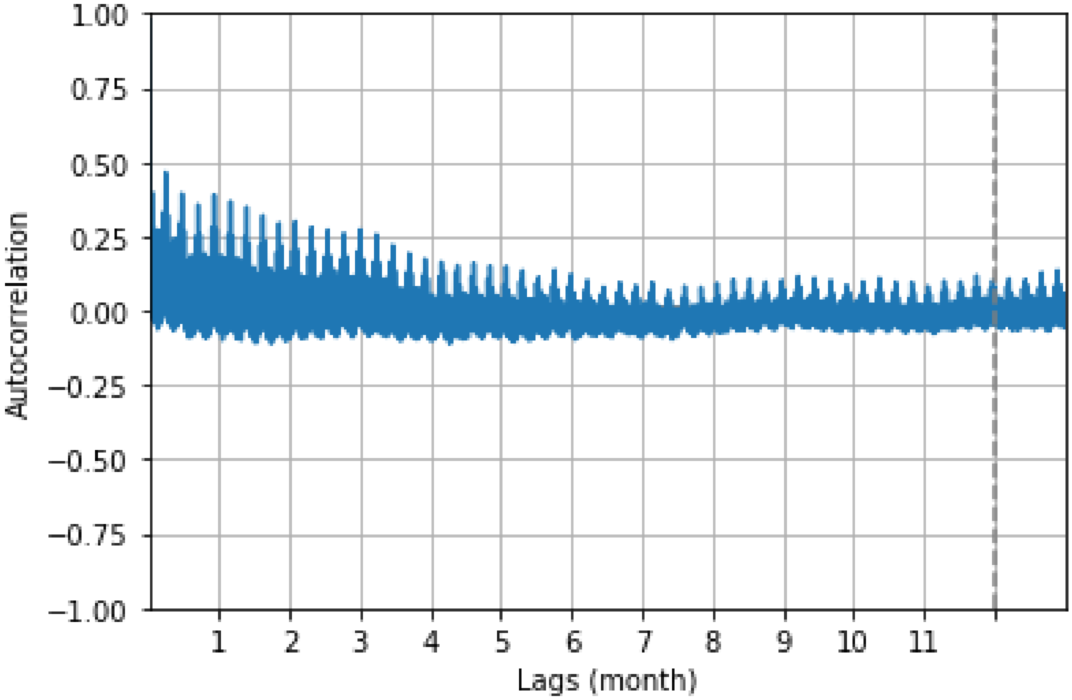


Figure 5.11: Autocorrelation of traffic showing no yearly seasonality

looser as months go by (see Figure 5.11). In other words, it is recommended to expect a certain pattern to last for only four weeks at most. Further examining its seasonality, it could also be observed it does not have a yearly seasonality as well, and its seasonality fades out as time passes by. In simple terms, the traffic pattern in January 2015 is not the same with January 2016. Rather, the traffic pattern in December 2015 is closer to the pattern of January 2016.

Given these seasonalities, we will explore how traffic changes in its short term and longer term.

Previous Day Traffic's daily seasonality could be perceived as causal, such that the traffic of yesterday could be attributed to the traffic of today. For instance, having an intense traffic yesterday morning could be linked with the incoming morning traffic of today. In inheriting the factors of previous days, nonetheless, we also need to take into consideration the concept of working and non-working day.

A *working day* refers to a day in which people are assigned on duty in an organization (J. Liu et al., 2008). For most organizations, it is defined to be on weekdays, Mondays to Fridays, while non-working days are on weekends, Sat-

Table 5.1: Comparison of traffic condition distribution between working and non-working

Traffic Condition	**Working Day**	**Non-working Day**
L	51.876%	76.973%
ML	4.051%	3.085%
M	31.917%	19.461%
MH	1.573%	0.180%
H	10.584%	0.301%

urdays to Sundays. Aside from weekends, though, non-working days also occur during holidays and government-announced class/work suspension. For our analysis, we treated those days as outliers to our data as the traffic pattern for that particular day is irregular compared with the other weekday working day records. These are important to consider as the transportation demand during working days are higher compared with non-working days (Z. Liu et al., 2016). For example, the transportation demand of Monday is significantly different from Sunday, thus referencing Sunday for the expected pattern of Monday would be erroneous.

Comparing the average traffic between working days and non-working days in one month, we could see a significant difference in terms of intensity and pattern (see Figure 5.12). Majority of intense traffic occurs at working days. Table 5.1 shows that heavy traffic consists 10.584% of the working day dataset, while only 0.301% of the non-working day dataset. Moreover, light traffic consists only 51.819% of the working day dataset while it accounts for 76.973% of the non-working day dataset. This, in terms of pattern, indicates that non-working day traffic may follow the peak hour-driven traffic pattern. Therefore, in defining a specific peak hour, we must consider non-working days as an inconsistent case if compared with working days.

Previous Week Traffic could also be related to its week-ago traffic pattern. There has been a concept known as Monday rush hour and Friday rush hour where traffic is expected to increase in the morning and at night respectively. Similar to the previous day approach, we also need to consider working days, but to a lesser degree. Instead, we would only take into account the holidays and suspensions which we will similarly treat as outliers in our reference. To support its weekly seasonality, in Figure **??**, it could be observed how closer the traffic pattern is with its one-week-ago pattern compared with its two-day-ago pattern.

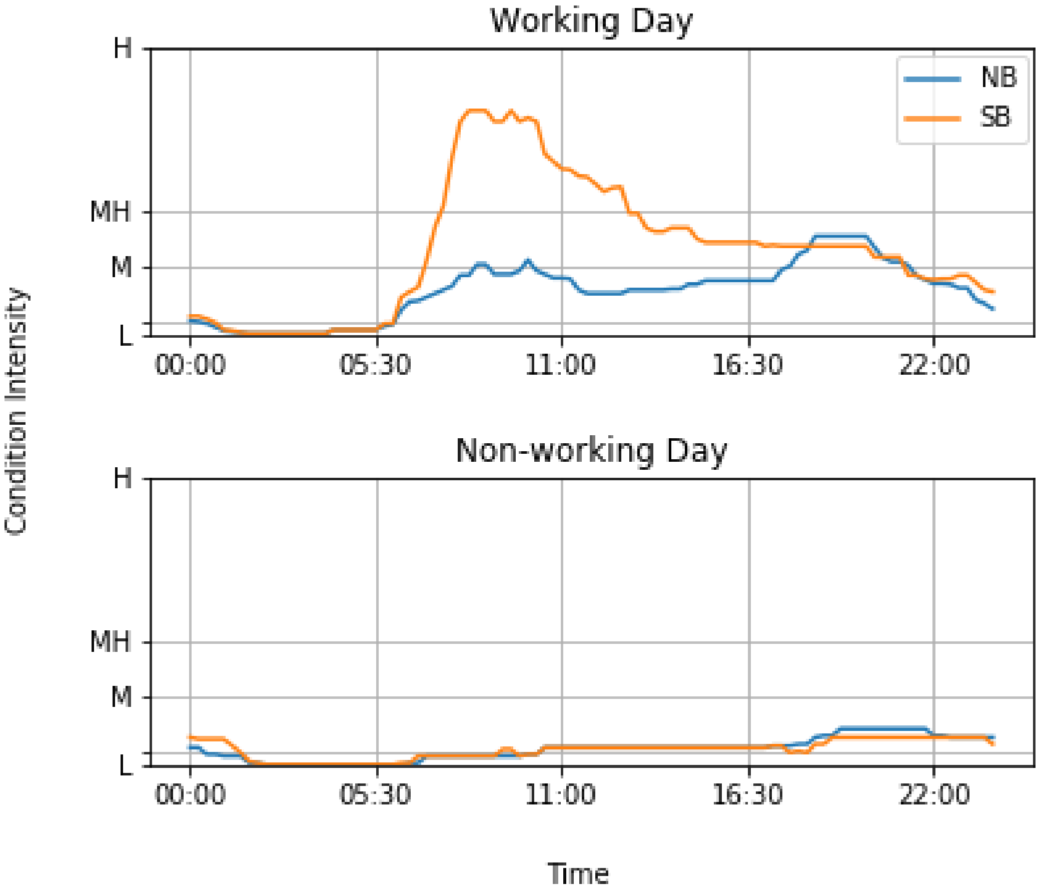


Figure 5.12: Comparison of traffic condition intensity between non-working and working revealing the lack of intense traffic for non-working days

Connected Road Segments

After analyzing the pattern of a road segment with its own pattern, we now analyze it as a part of a road. Aside from itself, its traffic could also be linked with its relationship with its connected road segments. Exploring the working day traffic of all road segments in Roxas Boulevard, it could be observed that there is an intensity relationship between these road segments. The intense traffic of Pablo Ocampo is carried over to Quirino in a similar intensity yet continuously decays as we go further to Anda Circle. Likewise, the intensity of the traffic of Blumentritt rises as we approach Antipolo then decays as we go further to Lerma.

Rolling and Expanding Features

The data only describes the traffic condition per timestep and the seasonality of traffic. Although traffic has pattern, there are instances when a certain disruption in traffic may cause congestion build up. Therefore, considering how the past traffic conditions may affect the current traffic condition can be considered as a factor for predicting traffic. Although the data from the previous timestep can be used as a reference in predicting the current traffic condition, the change in traffic in a 15-minute timeframe may not be significant. As seen in Figure 5.13, autocorrelation reveals that a traffic condition is highly likely to reoccur every 15 minutes with a correlation value of 0.9918. In other words, the traffic condition of the previous time step is highly likely to be the traffic condition of the current time step. However, this does not capture the effects of sudden outliers to the build up or decay of traffic. Thus, it does not fully describe how the past traffic conditions might affect the current traffic condition.

Generating rolling and expanding features based on a specific window size gives a bigger look as to what the current traffic might be based from the previous traffic conditions. Rolling and expanding traffic features such as the mean, the minimum, and the maximum with window size 4, 8, 24, 48, and 96 were generated from the original traffic data. To summarize the possible effects of sudden changes in traffic, the mean of the past traffic conditions based on a window size was generated. Figure 5.14 shows the standard deviation values of traffic for every window size. Standard deviation shows the variability or the distance of the data from the mean. From this, it can be observed that there is a logarithmic pattern in the growth of the standard deviation as the window size increases. The low standard deviation of smaller window sizes signifies that there is only a little to no change in traffic condition within that time frame. Meanwhile, large window sizes like window 96 which yield a relatively larger standard deviation value of 0.03531 than of window 4, implies that the traffic conditions in a 1-day time frame are more varied or are

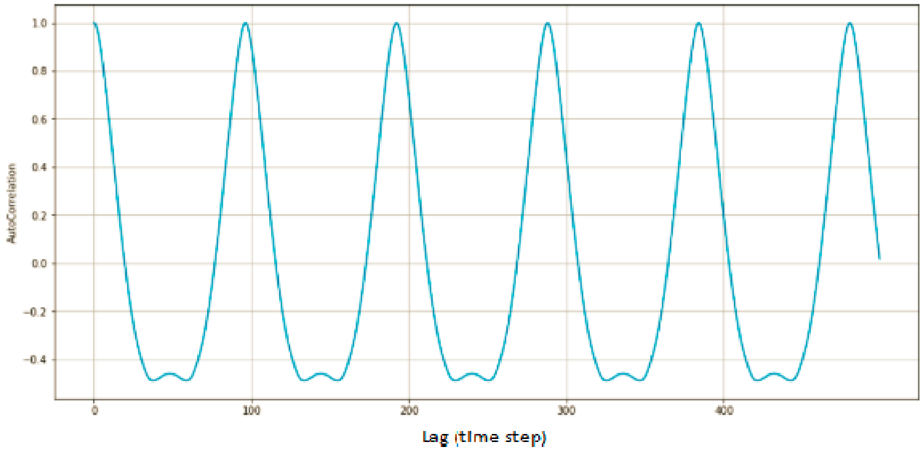


Figure 5.13: Autocorrelation of traffic for working days

more widely spread. With small window sizes, the values generated captures less of the trifle changes in traffic and gets more affected by outlier values which then results in a more generic information. However, as the window size increases, the effect of outlier values also get more neutralized because of the number of data considered, giving a broader summary of the past traffic.

Unlike the rolling mean, the expanding mean bases its computations on a growing window size and goes back to window size 1 when it reaches a specific maximum window size. This lessens the probability of the outliers getting majorly and consistently neutralized since it always goes back to the original traffic condition from the previous timestep. Thus, revealing less generalized details.

The minimum and maximum features, on the other hand, reveals the range of the dataset. Being sensitive to outliers, they provide outlier detection when compared to the average value of that specific set of data. A big difference between the values of the minimum and the maximum signifies a large progression in traffic condition. Although as the window size used gets bigger, it becomes harder to determine whether the sudden change in traffic condition occurred just a few timesteps before or if it is because of a farther instance which may have less to no effect to the current traffic condition.

Figures 5.15a and 5.15b show the average correlation of northbound road segments of Espana and southbound road segments of Roxas Boulevard per window size in relation to traffic for both rolling and expanding windows. On average, the original traffic has a strong relationship with rolling features having small window size, specifically window 4 with a correlation value of 0.7501. It is noticeable, however, that the strength of the relationship dwindles down as the window size increases. This reveals that although having a bigger window size means being

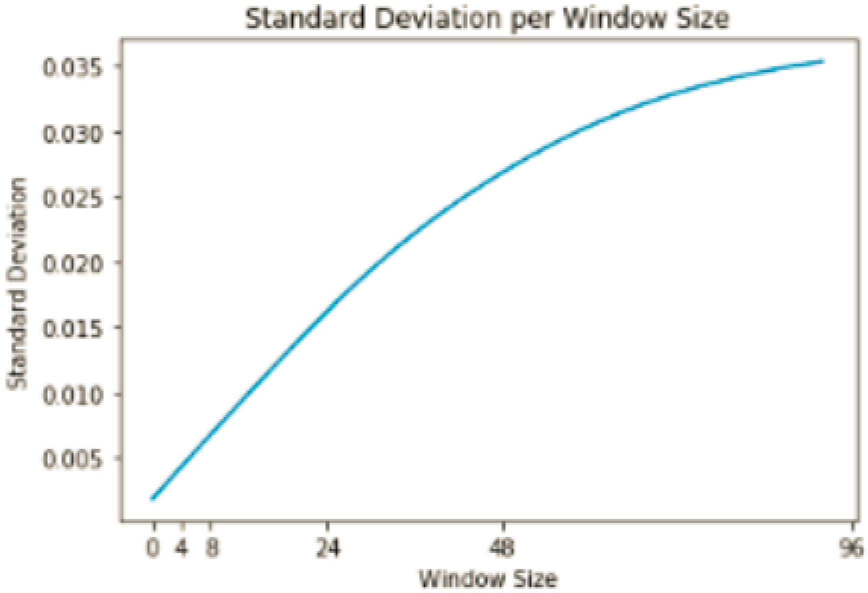
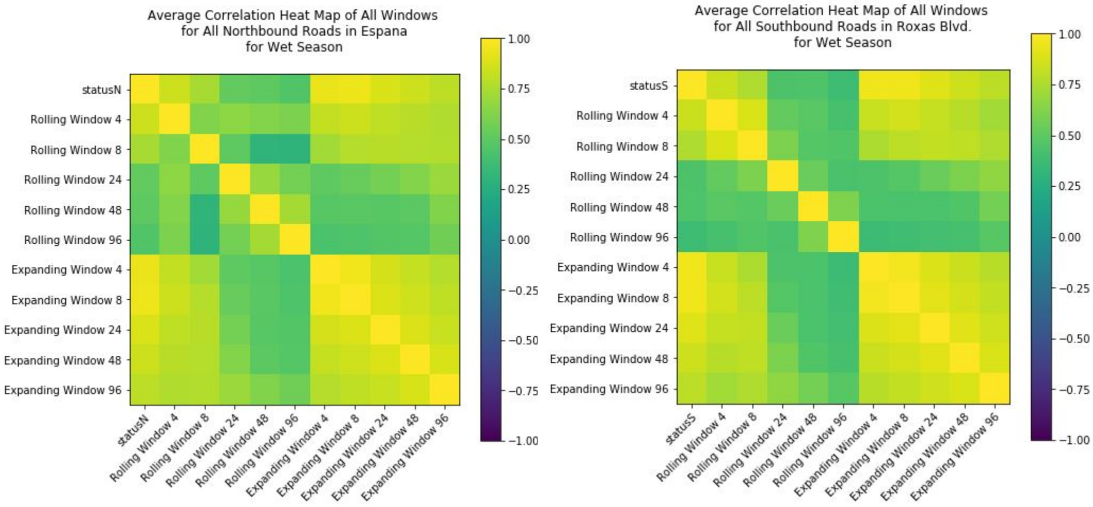


Figure 5.14: Standard deviation per window size of northbound traffic in Bluementritt for wet season

able to capture various changes in traffic condition, it does not give importance to the most recent traffic conditions. Big changes in traffic conditions that occurred way back and may not have any effect on the current traffic are being considered. Thus, causing its misalignment to the original data.

In the case of expanding windows, the strength of the relationship to the original traffic also decreases as the window size increases, albeit not so much as the rolling mean do. Looking into both graphs, the strength of the relationship given by window 4 for both rolling and expanding is distinctly stronger compared to the ones with larger window size. This might be because as mentioned earlier, traffic does not usually change significantly within a small window. Furthermore, a smaller window size means that its frequency of reverting back to the original value, making its generated value more accurate. Meanwhile, the farther from the past that it considers, the less it captures changes in traffic. This is shown in Figures 5.16a and 5.16b, where rolling and expanding at window 96 generalizes the high condition intensities into information that is no longer close to reality. Furthermore, the plodding decrease of correlation strength in expanding features as compared to rolling features is caused by limiting the number of windows that the feature grows to and considers and the restarting of its window size. Not limiting the window size growth of expanding features would produce values that are very far away from reality because it would consider every data from the previous days. It would contain values that consider data that are not relevant in

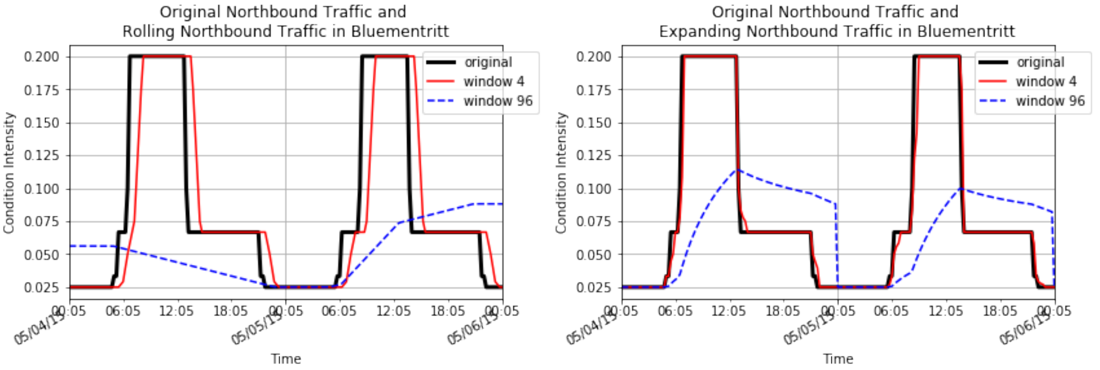


(a) EspanaRE

(b) RoxasRE

Figure 5.15: Average Correlation of Rolling and Expanding Features for Espana and Roxas Blvd.

predicting the future traffic.



(a) Original vs. Rolling

(b) Original vs. Expanding

Figure 5.16: Comparison of Rolling and Expanding windows 4 and 96 to original Southbound traffic in Bluementritt

5.1.3 Relationship of Weather to Traffic

Daytime and Nighttime

Time of Day

5.1.4 Effects of Weather to Traffic

Precipitation

Season by Precipitation Rate

Weather Disruption

5.2 Prediction Model Analysis

To test and evaluate the performance of the model in the presence of weather changes, two datasets for both dry (January to April, and November to December) and wet season (May to October) were used. Traffic and weather data were provided by MMDA and WWO, respectively. The models were evaluated using root mean squared error (RMSE) and mean absolute error (MAE).

5.2.1 Traffic-Only Model

To evaluate the performance of TOM, and the effectiveness of engineering features for traffic, combination of features were used and compared for all road segments for both wet and dry season. These features include temporal information about the current traffic (i.e. month, day, hour, minute, and day of the week), past traffic of 96 intervals in the past or traffic the day before, flags for work days and peak hours, and rolling and expanding window of traffic 15 minutes in the past. Based on earlier discussions, rolling and expanding windows are not correlated with each other. Thus, rolling and expanding window features are used together in the model. In further figures in discussing the Traffic-Only Model, the different feature combinations are denoted as the following:

1. *OT* - Features that only consist of the past traffic a day before the current;
2. *OTWP* - Features including past traffic a day before the current, and flags on the workday and peak hour, and;
3. *WPRE*- Features including OTWP, plus the addition of rolling and expanding features of the traffic 15 minute in the past of the current. In subsequent discussions, any number following this notion pertains to the window size of both the rolling and expanding features (e.g. WPRE 4 pertains to the features that include rolling and expanding features of window size 4).

As seen in Figure 5.17 the inclusion of the information about working days and peak hours for all road segments improved the performance of the model by only 22% having originally an RMSE of 0.167 which improved to 0.113. As discussed earlier on the trend and patterns of traffic, though there is a difference between the trends of working days and non-working days, moderate traffic is most frequent in both trends in the majority of the road segments, This implies that the mean traffic intensity for both working and non-working days are similar. This similarity attributes to the small improvement in the prediction. Furthermore, including rolling and expanding window features for traffic significantly improved the prediction by around 53%, from having an RMSE of 0.129 to 0.061 after adding the said features. Rolling window features present a generalized information regarding the trend of the immediate past traffic. As discussed earlier, average traffic does not change significantly within a small window.

To evaluate if the model can successfully identify the trends present for each road segment, and its ability to predict its traffic, the performance of the model using OTWP or Original Traffic with the addition of work day and peak hours was observed. Results of this evaluation are illustrated in Figure 5.18. It is noticeable

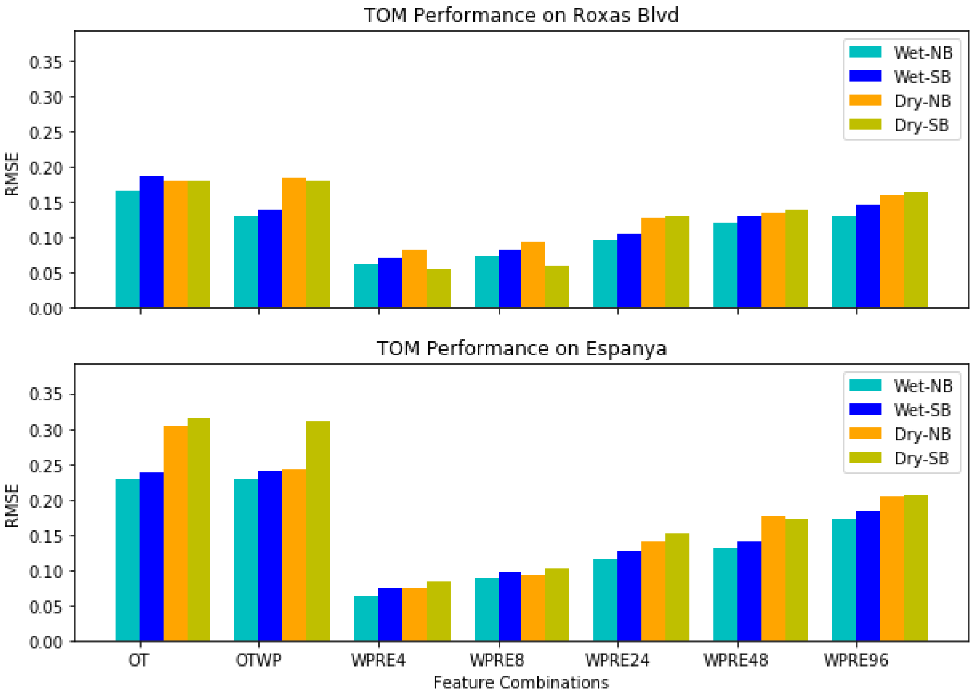


Figure 5.17: Comparison between performance of TOM on Roxas Boulevard and Espana using different feature combinations

that the model predicted the southbound traffic of Espanya during the wet season quite well, having a mean RMSE of 0.095 compared to the other. However, this is because of the diversity of the southbound traffic of Espanya during the wet season, having a Moderate traffic congestion majority of the time. Given only data of the previous day's traffic, and information on work day and peak hours, the model cannot easily model the sudden peaks and traffic reports on Heavy traffic. Additionally, given the percentage of the report on Heavy traffic congestion of the data, having only 10.584% during the working days, and 0.301% during the non-working days, the model lacks knowledge on modeling expected heavy traffic.

5.2.2 Weather-Only Model

In earlier discussions regarding the relationship between weather variables with traffic, most weather variables had a weak relationship with traffic. To further evaluate this, the Weather Only Model was tested using three different combinations of weather variables which are:

1. Original Weather (OW) - all weather variables

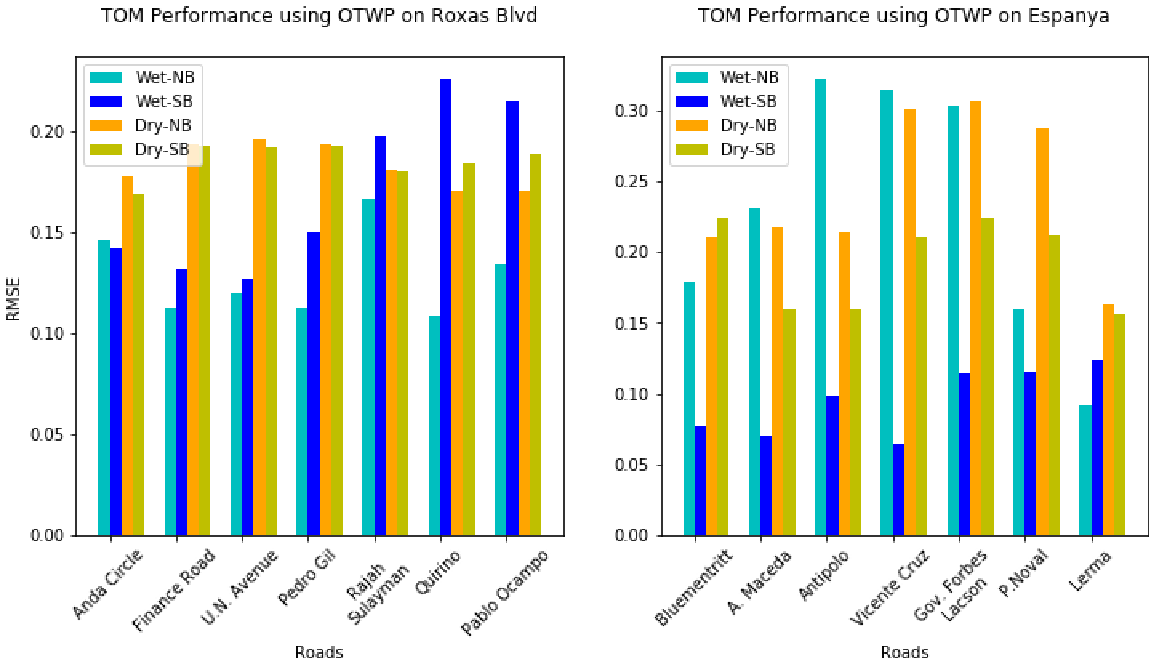


Figure 5.18: Performance of TOM on Roxas Boulevard and Espanya using OTWP traffic feature combination

2. All Correlated Weather (ACW) - all weather variables that have correlation between 0.1 to 0.3

3. Correlated Weather (CW) - all correlated weather variables excluding the redundant ones

Given the weak relationship between weather variables with traffic, the model could only predict traffic about 80 to 84% after using weather variables for input as compared to using past traffic that can predict traffic by 90%. Furthermore, the model's performance did not improve significantly with different combinations (see Figure 5.19) of weather variables having changes in error ranging from only 0.001 to 0.003.

Illustrated in Figure 5.19 is the performance of the weather only model using the three feature combinations for the road segment of Pablo Ocampo and Blumentritt. As seen in the figure, the model predicted the northbound traffic in Pablo Ocampo better than the southbound traffic. According to data analysis performed above, the northbound traffic in Pablo Ocampo has more reports on heavy traffic than the southbound. Likewise, the model predicted the southbound of Bluementritt better than the northbound. However, in Pablo Ocampo, the

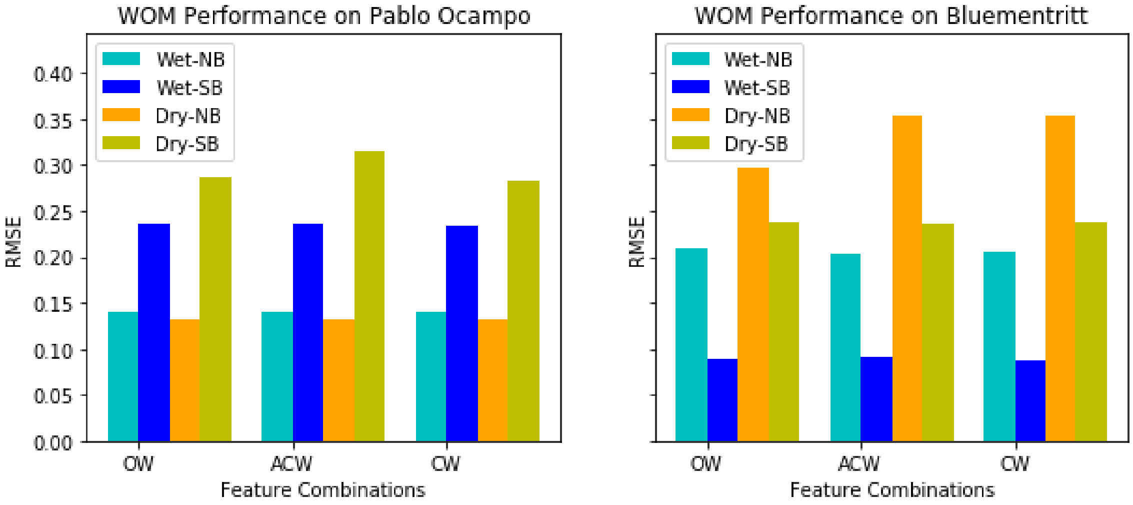


Figure 5.19: Performance of WOM on Pablo Ocampo and Bluementritt using weather feature combinations

model performed better using the northbound traffic for both dry and wet season, as for Bluementritt, the model performed better when using the wet season dataset, for both northbound and southbound. During the wet season of 2015, the traffic in Bluementritt was less diverse having a mean traffic for southbound of 0.187 and mean traffic for northbound of 0.208, both values close near the value for moderate traffic of 0.238. This means that traffic congestion level in Bluementritt for both southbound and northbound during the wet season is mostly moderate. This is further supported by evaluating the model's performance with other connected road segments as seen in Figure 5.20 which shows the performance of the weather only model using all weather features for all road segments. Other road segments follow the said behavior.

5.2.3 Fusion Analysis

Different fusion techniques and approaches were used and evaluated. The performance of fusing in the feature and decision level was first evaluated. Then, the performance of fusing in the decision level using a number of fusion algorithms was observed. Table 5.2 displays the different performances in different fusion approaches and techniques.

Although the results display a difference between the performance of fusing at the feature level to decision level, the model only improved by 6.87% in fusing at the decision level.

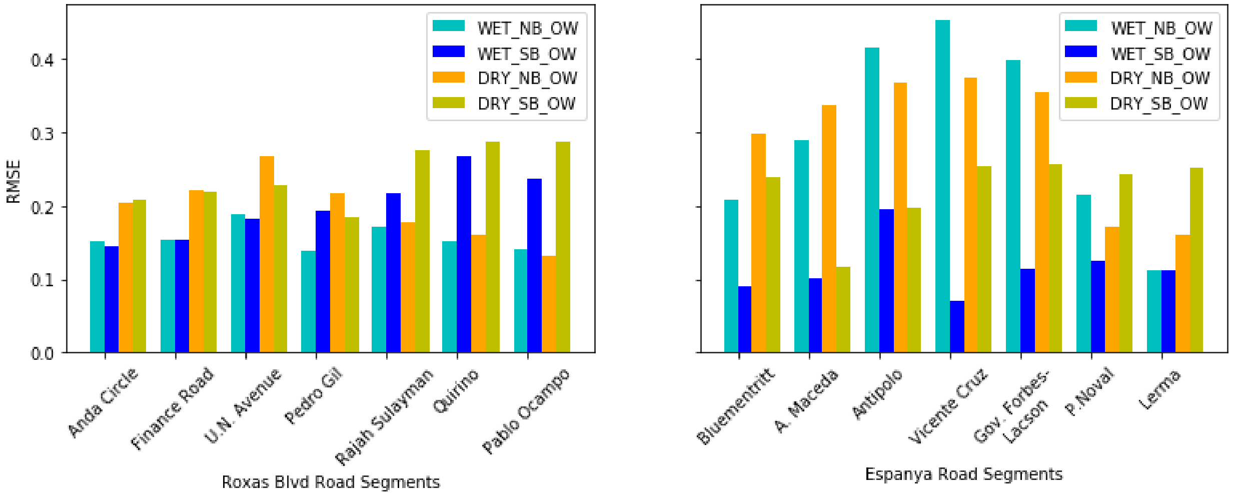


Figure 5.20: Performance of WOM using OW weather feature combination for all road segments

Fusion Techniques	**OTWP + OW**		**WPRE 4 + OW**		**WPRE 96 + OW**	
	RMSE	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**
DBN Feature Fusion	0.262	0.156	0.079	0.030	0.219	0.122
DBN Decision Fusion	0.244	0.173	**0.070**	**0.025**	0.144	0.079
RNN Decision Fusion	**0.068**	**0.054**	0.171	0.131	**0.103**	**0.085**
WA Decision Fusion	1.087	1.048	0.497	0.457	0.727	0.700

Table 5.2: Performance of different fusion techniques in different feature combinations

The performance of WA in fusing data with the least-square estimate was greatly outshone by the two neural network fusion algorithms. Fusing with WA gained an RMSE of 1.087 and an MAE of 1.048, having a significant difference in error of 1.019 to 0.843 as compared to the other fusion algorithms. The algorithm for weighted average for fusing the predictions made used of least square estimate method to determine the fusion's weights. Least-square estimate is defined as the minimum of the sum of squared residuals, or the difference between the actual and the predicted. This method is most useful when used in data which are more linear, where the pattern of the data can be extracted through simply just getting the residual, the slope of the line, and the coefficients of the input. This method of pattern extraction is called regression. The coefficients of the input generated after regression analysis are used as weights in the study's Weighted Average algorithm. Given the behavior of the algorithm, it fails to extract patterns from the non-linear data. For the subsequent experiment and analysis, WA is disregarded.

RNN outperforms DBN, and other fusion algorithms by 74%. RNN is implemented as an LSTM, or a long short-term memory, compared to DBN which is implemented as a regressor and a classifier. Because of RNN's power of long short-term memory, it can use their internal memory to process a sequence of inputs, successfully extracting ordered patterns.

In evaluating the effect of the inclusion of weather as a factor in predicting traffic, the performance of the Traffic-Only Model, the model that can predict traffic using only the immediate past traffic, is compared with the performance of the fusion for all techniques. Results are illustrated in Figure 5.22, which displays that the prediction improves after using weather as a factor. Though the improvement in performance is only 14%, the improvement was consistent for different combinations of features. A contributor to the low improvement is the number of observations that weather is present when traffic least expected. For example, according to the weather data, heavy rainfall was present during September 11-12 of 2015, a Friday and a Saturday, when traffic is least expected. Additionally, rain started to fall at night of September 11, extending until the early morning of September 12. There were few records that successfully capture the contributing factor of weather towards traffic given the time periods disrupting weather variables were often present. For visualization purposes, the predicted traffic is illustrated in Figure 5.21

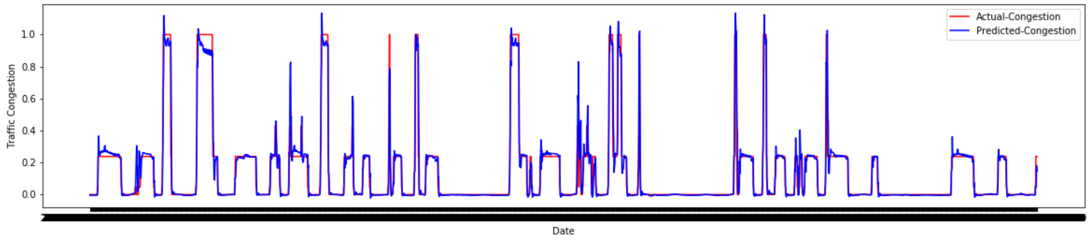


Figure 5.21: Final Prediction of Southbound of Pablo Ocampo for the Month of September

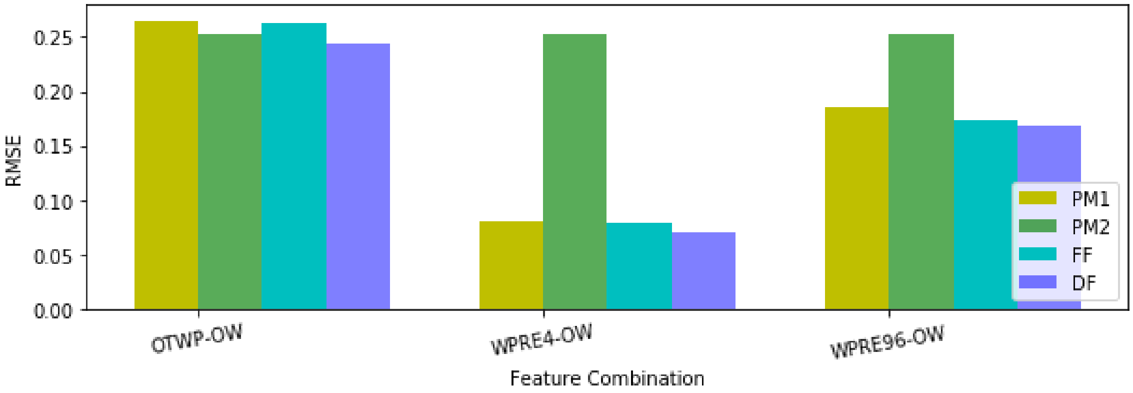


Figure 5.22: Comparison of DBN model in predicting the southbound of Pablo Ocampo for the wet season

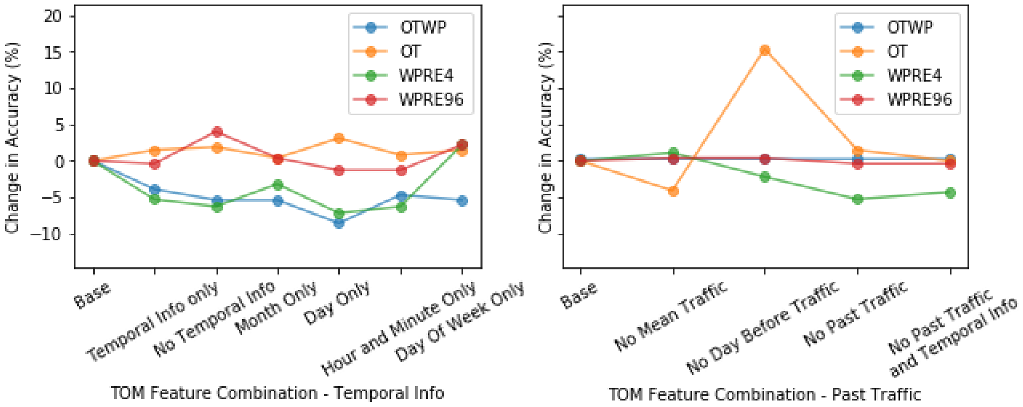


Figure 5.23: Sensitivity of TOM with different feature combinations

5.2.4 Sensitivity Analysis

Traffic-Only Model

First, the sensitivity of the predictions models with changing input variables was evaluated. In evaluating TOM, the different input variables were removed per experiment to see the effect of its inclusion. The combination of input variables were done in the feature combination in earlier evaluations of the model (e.g. OTWP, OT, WPRE, OW, etc). For example, variables of the OTWP feature combination were tinkered. Then, the variables of OT combination were evaluated, and so on. The temporal information was the first set of features of the combination that were removed one-by-one, then information on past traffic. The base model, with all the features including temporal and past traffic, was compared with the model with the removed variables (with or without temporal information or past traffic), and its change in performance was evaluated. The changes in performance illustrates the sensitivity of the model.

Evaluation on the sensitivity of TOM is illustrated in 5.23. It has been discussed in the early sections in evaluating TOM that the inclusion of Work day, and Peak hour variables result to a 22% improvement in performance. The removal of temporal information decreased the performance from 3.9 to 8.6% when used with work day and peak hour. Not one temporal information variable increase the accuracy of the base which used all the temporal information, and all past traffic features. However, without the work day and peak hour, removal of the temporal information seems to have increase the performance, but only by 1 to 3%. Results show that temporal information, when used with working day and peak hour variables without past traffic information, decreases the performance

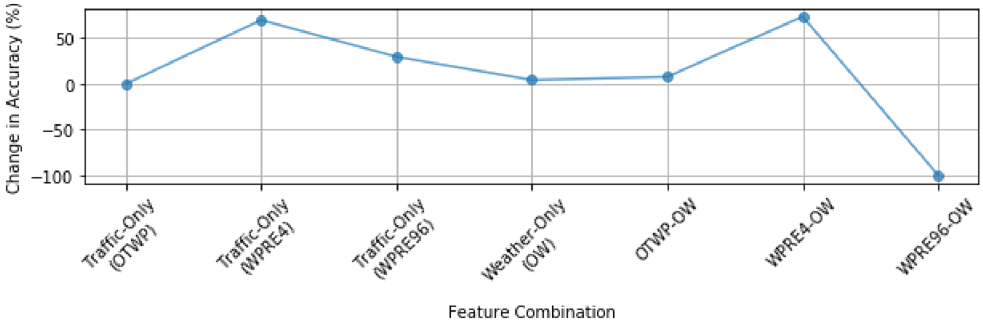


Figure 5.24: Change in performance in different feature combinations

by 4%. Moreover, each variable excluding the *Day* variable are independent of each other as seen by the change of accuracy when using only Month, Hour and minute, or day of week. Using the day variable without the support of other temporal information have decreased the performance by 8.6%. Results show that traffic does not have any pattern with the *Day* variable. As for the use of temporal information without working day and peak hour, the model performed better. Working day and peak hour variables show a generalized information of traffic at a specific time period. Without this information, the temporal information instead gives information on the traffic such as what hours, or day of week are there heavy traffic condition intensity.

Looking into the effect of the past traffic feature when used with the work day and peak hour, using either the mean of traffic 2 weeks ago, and a day before did not affect the performance, only having differences of 0.03%. However, There are significant change when work day and peak hours are not used. In using only the traffic a day before, the performance decreased by 4.1%. Using the traffic of 2 weeks ago significantly increased the performance by 15.4%. Results show that the base model for OT that used all the past traffic features of a day and 2 weeks ago and all temporal information without workday and peak hour, gets most of its prediction bases from the general traffic 2 weeks ago. This shows that this variable have great importance in predicting traffic, when working day and peak hours are not used.

Decision Fusion Model

The base model for evaluating the fusion between using the different prediction models, and fusion models is the Traffic-Only for OTWP feature combination. As discussed in the Fusion Analysis, considering the weather prediction in fusing at decision level resulted in a 14% improvement in performance. However, in evaluating the performance of the weather in predicting traffic, weather only contributed about 27% of the prediction. As mentioned in earlier discussions, traffic data mostly comprises of Moderate traffic condition intensity reports. Additionally, periods where there is a disruption in weather in which traffic is expected lies in weekends, when traffic is least expected. The data does not completely represent the relationship between weather and traffic, thus resulting to the small weight in prediction.

Chapter 6

Conclusion

Urban centers have long-since been fighting the natural occurrence of traffic congestion in the past decades. To help solve this problem, several traffic models were already developed while also considering weather as an effect. However, these models were not designed and tested in places with substandard drainage systems and poor road management. This research aims to adapt a weather-aware traffic model to cater these places. In this paper, analysis on the patterns of traffic and weather data were explored through correlation analysis. Features were also extracted to gain more insights as to what variables have a relationship with traffic congestion. An existing traffic model using DBNs and data fusion techniques was adapted to predict the traffic congestion in selected roads in Manila. Different experiment settings were done to show how different traffic and weather features affect the traffic prediction.

Analysis show that patterns and trends are still evident even in urban centers where disruptions in traffic and weather are present. Additionally, extracting information from the categorical data of traffic congestion is possible and can be quite insightful. In evaluating the model in predicting future traffic, results show that using the immediate past traffic is insightful and bears importance in decision making. Results also show that fusing at decision-level improves accuracy in prediction better than feature-level. Moreover, though DBNs bear the ability to extract patterns, RNNs outperforms DBN in extracting the patterns of time-series data.

Chapter 7

Future Works

Using historical data is a good and common entry for data-driven researches. However, in the case of traffic, it is better to consider real-time information to predict future traffic conditions. In the event that something unexpected happens, such as road accidents or just-closed lanes, the historical traffic data may not be as useful as it is. Sudden events may also include typhoon warnings, which has no annual pattern and cannot be be easily predicted. With historical traffic and weather data combined with real-time information, future traffic predictions may become more accurate.

Considering traffic reports in social media may also be an additional feature in the model. Social media platforms such as Twitter and Facebook allows people to let out their frustrations including everyday problems such as road traffic. One study (Gu et al., 2016) retrieves real-time traffic information by crawling, processing, and filtering public tweets to detect road traffic incidents using natural processing language techniques. Using this as an added feature in the model may boost its accuracy.

Unpredicted road catastrophes also contribute to traffic congestion, including road accidents, road constructions, and recently-closed lanes. Car accidents block the roads, which then hinder other vehicles to efficiently pass through, causing heavy traffic jam (Wang et al., 2009). Road constructions such as building of bridges, expanding the roads, fixing the water pipes underneath the roads, among others, also tend to increase traffic congestion, especially if there is no notice beforehand. The sudden closing of a road may also contribute to traffic jams, since it disrupts the normal flow of traffic. Studying these kinds of road catastrophes may be helpful in achieving a better traffic prediction. Data may be collected as historical data or using real-time information.

In urban places where drainage systems are not properly managed or road infrastructures are not properly built and maintained, the amount of rain accumulating on the roads may increase traffic congestion as vehicles have difficulty moving forward. It is worse on places where adverse weather conditions such as typhoons bring heavy rains, which then causes immense flooding if the rain is not properly drained. Although the existing application Waze allows drivers and passengers to report flooding incidents in the roads, Waze does not use these reports in learning and predicting future traffic conditions. In future works, considering the height of flood as a contributing factor may help the model in predicting future traffic condition.

Appendix A

Research Ethics Documents

This appendix contains all documents related to research ethics.

DE LA SALLE UNIVERSITY
General Research Ethics Checklist

This checklist is to ensure that the research conducted by the faculty members and students of De La Sall University is carried out according to the guiding principles outlined in the Code of Research Ethics of th University. The investigator is advised to refer to the <u>De La Salle University Code of Research Ethics and Guid to Responsible Conduct of Research</u> before completing this checklist. Statements pertinent to ethical issues i research should be addressed below. The checklist will help the researchers and evaluators determine whethe procedures should be undertaken during the course of the research to maintain ethical standards. Th University's <u>Guide to the Responsible Conduct of Research</u> provides details on these appropriate procedures.

Details of the Research	
Students	MAGPALE, Nicolle G. NIEVA, Dyan Raisa L. REAMON, David Angelo H. RECCION, Maria Victoria B.
Thesis Adviser	CABREDO, Rafael A., Ph.D.
Department	Software Technology Department
Title of the Research	TrafficBaTo: Towards a Weather-Aware Traffic Model
Term(s) and Academic year in which research is to be conducted	Terms 1 to 3 AY 2017-2018

This checklist must be completed AFTER the* De La Salle University Code of Ethics *has been read and BEFORE gathering data.

Questions	Yes	No
1. Does your research involve human participants (this includes new data gathered or using pre-existing data)? If your answer is **yes**, please answer **Checklist A (Human Participants).**		✓
2. Does your research involve animals (non-human subjects)? If your answer is **yes**, please answer **Checklist B (Animal Subjects).**		✓
3. Does your research involve Wildlife? If your answer is **yes**, please answer **Checklist C (Wildlife)**.		✓
4. Does your research involve microorganisms that are infectious, disease causing or harmful to health? If your answer is **yes**, please answer **Checklist D (Infectious Agents).**		✓
5. Does your research involve toxic/chemicals/ substances/materials? If your answer is **yes**, please answer **Checklist E (Toxic Agents).**		✓

Research with Ethical Issues to address:

If you have a YES answer to any of the above categories, you will be required to complete a detailed checklist for that particular category. A YES answer does not mean the disapproval of your research proposal. By providing you with a more detailed checklist, we ensure that the ethical concerns are identified so these can be addressed in adherence to the University Code of Ethics.

Declaration of Conflict of Interest

[✓] I do not have a conflict of interest in any form (personal, financial, proprietary, or professional) with the sponsor/grant-giving organization, the study, the co-investigators/personnel, or the site.

[ ] I have a personal/family or professional interest in the results of the study (family members who are co-proponents or personnel in the study, membership in relevant professional associations/organizations).

Please describe the personal/family or professional interest: ____________________

____

[ ] I have propriety interest vested in this proposal (with the intent to apply for a patent, trademark, copyright, or license)

Please describe propriety interest: ____________________

____

[ ] I have significant financial interest vested in this proposal (remuneration that exceeds P250,000.00 each year or equity interest in the form of stock, stock options or other ownership interests).

Please describe financial interest: ____________________

____

Declaration

We certify that we have read and understand the De La Salle University Code for the Responsib Conduct of Research and will abide by the ethical principles in this document. We will submit a fin report of the proposed study to the DLSU-Research Ethics Office. We will not commence with da collection until we receive an ethics review approval from the College Research Ethics Committee.

MAGPALE, Nicolle G.
Name and Signature of Student 1

NIEVA, Dyan Raisa L.
Name and Signature of Student 2

REAMON, David Angelo H.
Name and Signature of Student 3

RECCION, Maria Victoria B.
Name and Signature of Student 4

Endorsement from thesis adviser to the thesis panel for proposal defense...

CABREDO, Rafael A., Ph.D.
Name and Signature of Adviser

Nov. 3, 2017
Date

Endorsement from thesis adviser to the thesis panel for final defense...

This is to certify that the research was conducted in a manner that adheres to ethical research standard I am thus endorsing the group for final defense.

CABREDO, Rafael A., Ph.D.
Name and Signature of Adviser

____
Date

RESEARCH ETHICS CLEARANCE FORM
For Thesis Proposals[1]

Names of student researcher/s :	*MAGPALE, Nicolle G.* *NIEVA, Dyan Raisa L.* *REAMON, David Angelo H.* *RECCION, Maria Victoria B.*
College:	College of Computer Studies
Department:	Software Technology Department
Course:	BS Computer Science with specialization in Software Technology
Expected duration of project:	from: T1 AY '17-'18 to: T3 AY '17-'18

Ethical considerations

The data to be used is publicly available data.

To the best of our knowledge, the ethical issues listed above have been addressed in the research.

CABREDO, Rafael A., Ph.D.

Name and signature of adviser/mentor

Date: 12/13/2017

LAO, Angelyn R., Ph.D.

Name and signature of panelist

Date: 12/12/2017

DELOS SANTOS, Duke Danielle T.

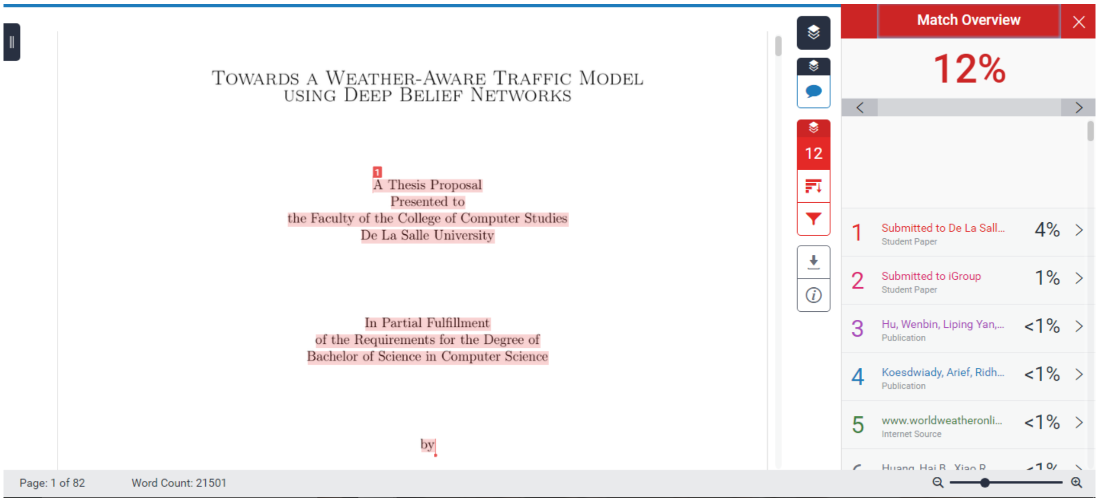
Name and signature of panelist

Date: 12/12/17

Appendix B

Turnitin Similarity Report

This appendix contains the similarity report from Turnitin.



TOWARDS A WEATHER-AWARE TRAFFIC MODEL
USING DEEP BELIEF NETWORKS

1
A Thesis Proposal
Presented to
the Faculty of the College of Computer Studies
De La Salle University

In Partial Fulfillment
of the Requirements for the Degree of
Bachelor of Science in Computer Science

by

Match Overview

12%

1	Submitted to De La Sall... Student Paper	4%
2	Submitted to iGroup Student Paper	1%
3	Hu, Wenbin, Liping Yan,... Publication	<1%
4	Koesdwiady, Arief, Ridh... Publication	<1%
5	www.worldweatheronli... Internet Source	<1%
6	Huang, Hai B., Xiao R...	<1%

Page: 1 of 82 Word Count: 21501

Figure B.1: Turnitin Similarity Report

Appendix C

Research Project Timeline

This appendix contains the time line of the research project.

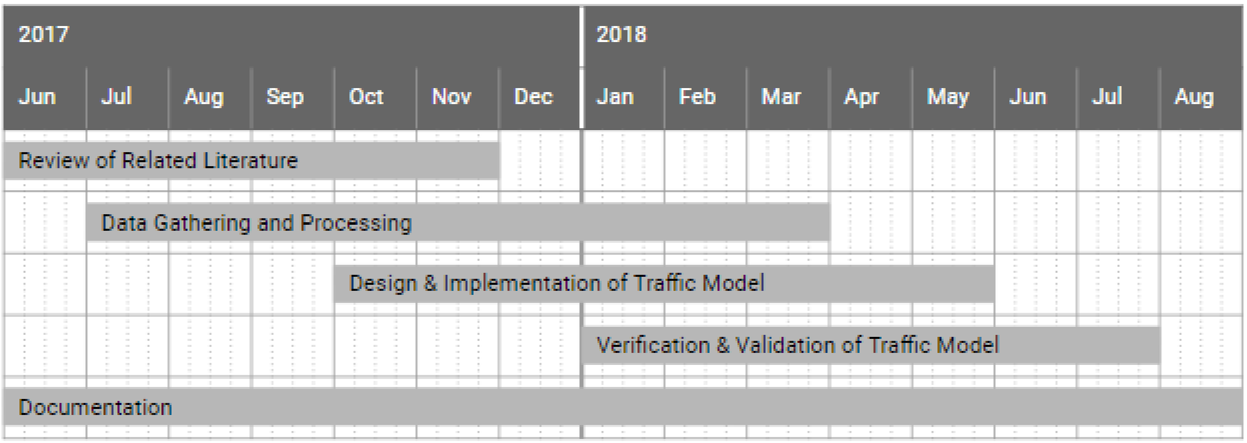


Figure C.1: Research Project Timeline

Appendix D

Preliminary Results

In this section, an implementation of feature fusion, using the collected traffic and weather dataset, will be discussed through a feedforward ANN. To evaluate the performance of feature fusion, two experiments will be performed: predicting the traffic condition using weather variables only and predicting the traffic condition by applying feature fusion to both the traffic condition and weather variables.

D.1 Data Pre-processing

Before feeding the data to the ANN model, the data must be cleaned and processed. Recall that the collected dataset for both the weather and traffic are sampled at a 15-minute interval and an hourly interval respectively and that the traffic dataset contains missing records (see Section 4.1). Given these anomalies, interpolation must be done for the traffic dataset to supply the missing records and intervals, whereas resampling and interpolation must be performed for the weather dataset to match the traffic dataset's interval.

D.1.1 Traffic Dataset

For the traffic dataset, we must first separate the records based on their respective road segments. For instance, we must have a separate dataset only containing the records from Taft Avenue, Quirino, Buendia and the other road segments. This is to ensure that our interpolation later on will not be influenced by other road segments rather than itself.

Before cleaning the data, we must consider that interpolation requires numerical values so that the missing values can be estimated. Therefore, we must convert the traffic conditions for both the northbound and southbound to their numerical equivalents. Since they are classified as light, moderate and heavy, we could rank them based on their intensities. As a result, light, moderate and heavy are classified into 0.0, 0.5 and 1.0 respectively. These values are chosen so that the min-max scaling normalization is no longer needed to perform later on.

Next, to clean the traffic dataset, we have to consider the two cases of missing data that it has: having *none*(N) conditions and having a particular interval not recorded at all. To resolve the first case, first, we must remove all *none*(N) condition from both the northbound and southbound traffic condition features. Next, we perform linear interpolation to fill in these filtered records. For the second case, we must fill in the unrecorded intervals. Hence, although the traffic dataset is already in our target series interval, we need to resample the dataset at a 15-minute interval so that all the intervals, including the missing ones, will be added. Next, we perform linear interpolation again to fill in these missing records.

D.1.2 Weather Dataset

Before processing our dataset, recall that our weather dataset contains 20 weather variables, having the temperature, heat index, dew point, wind chill and feels like on both their Celcius and Fahrenheit form, and wind speed and wind gust on both their miles per hour and kilometers per hour form. In our case, we opted to use their Celcius form and kilometers per hour form. Thus, scoping down our weather variables to 13.

For the weather dataset, since, fortunately, our data is complete per interval, we could directly start resampling our records. But, before this, we must ensure that our weather variables are in their numerical equivalents as a preparation for interpolation later on. In the case of our dataset, only the weather condition is in its categorical form. To rank the weather condition, we adapted Tse (2003) approach of ranking weather conditions based on its attenuation of incoming solar radiation (see Section **??**). With that, the weather condition data is ready for interpolation and is already normalized.

After processing our dataset, we could now match our weather dataset to the intervals of our traffic dataset. To do this, we must first resample our dataset at a 15-minute interval. Next, we perform linear interpolation to fill in the missing intervals, similar to how we filled the missing records from our traffic data.

D.2 Input Dataset

To scale down our experiment, we sampled three consecutive months of 2015 as our dataset (from September 2015 to November 2015) for both traffic and weather. Furthermore, we only used the traffic records of Taft Avenue with the weather of Manila as our input dataset. For the features, we only considered the southbound traffic condition for the traffic. On the other hand, we considered 13 features for the weather. These include temperature (in Celsius), wind speed (in kilometers per hour), weather condition, precipitation (in millimeters), humidity, visibility, pressure, cloud cover, heat index (in Celsius), dew point (in Celsius), wind chill (in Celsius), wind gust (in kilometers per hour), and feels-like (in Celsius). It must be noted, however, that the correlation of these features has not been considered in this experiment.

D.3 Developing the Model

Developing our feedforward ANN consists of 5 steps: preparing the data, defining the training and test data, building the ANN model itself, training the model and finally evaluating the model. Our model will be implemented using Python, utilizing the Keras library with Tensorflow. For our experiment, we would be performing two cases: predicting the traffic condition using weather variables only and predicting the traffic condition by applying feature fusion to both the traffic condition and weather variables.

D.3.1 Preparing the Data

Aside from the pre-processing that was performed earlier, we also have to perform some additional pre-processing targeted for our model. First, we have to merge both our traffic dataset and weather dataset. Since they already have matching intervals and the traffic dataset is already separated based on its road segments, we could simply merge it without any additional processing involved. It is important to note that only one road segment dataset will only be used per model, especially in our case where the weather dataset is the generalized weather for all road segments, since the generalized weather may be biased on a particular road segment.

Next, we remove the unnecessary features from our merged dataset. These include the road, road segment and the date and time, leaving us with the numerical traffic features and weather variable features. Finally, we must perform normalization to standardize the values. In our case, we performed min-max scaling normalization (see Section **??**) to scale our data from 0.0 to 1.0.

D.3.2 Defining the Training and Testing Dataset

The dataset is split into two sub-datasets: the training dataset which includes September 2015 to October 2015 and the testing dataset which includes November 2015. From this, the target label, the feature that we want to predict, and the training features, the features from which the prediction will be based on, are defined. To further minimize the scope of the experiment, we would only predict the traffic condition of the southbound lane. Thus, for the experiment, our target label would be the traffic condition of the southbound lane.

In order to illustrate the benefits of feature fusion, two experiments will be performed. First, the training features will be solely based on weather variable features. Second, the training features will include the traffic condition from the southbound lane and the weather variable features thus, performing feature fusion on the traffic and weather variables features. To further illustrate these experiments, Figures D.1 and D.2 show the diagram of these scenarios.

Figure D.1: Predict Southbound Traffic without Feature Fusion

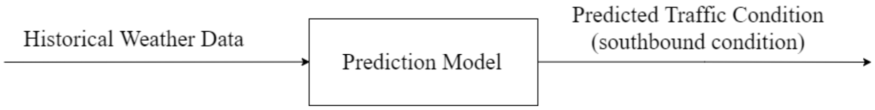
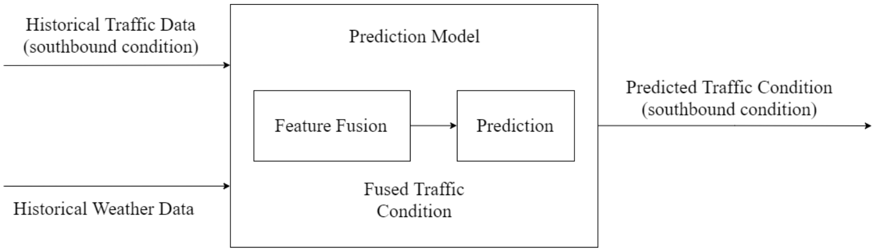


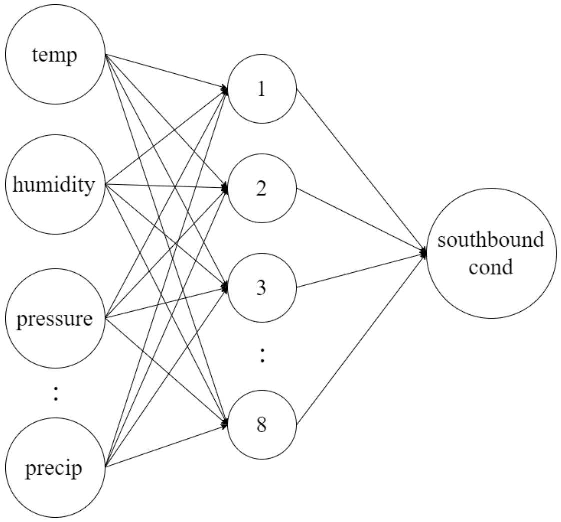
Figure D.2: Predict Southbound Traffic with Feature Fusion



D.3.3 Building the Model

In building our feedforward ANN model, we used three densely-connected neural network layer: one for the input, one for the hidden layer and one for the output. For each of our experiments, both uses rectified linear unit (ReLU) as the activation function, and each has 13 input dimensions and 14 input dimensions respectively. The dimensions correspond to the number of training features that we will use for the model. The model has 13 weather variable features for our first experiment scenario, whereas we have 13 weather variable features and 1 traffic feature for our second scenario. For both experiments, it uses one hidden layer that similarly uses ReLU as its activation function, and has 8 input dimensions. It also has one output layer with only one output dimension for our southbound traffic condition. Figure D.3 and D.4 illustrate the diagram of these models.

Figure D.3: ANN Model for Predicting Southbound Traffic without Feature Fusion



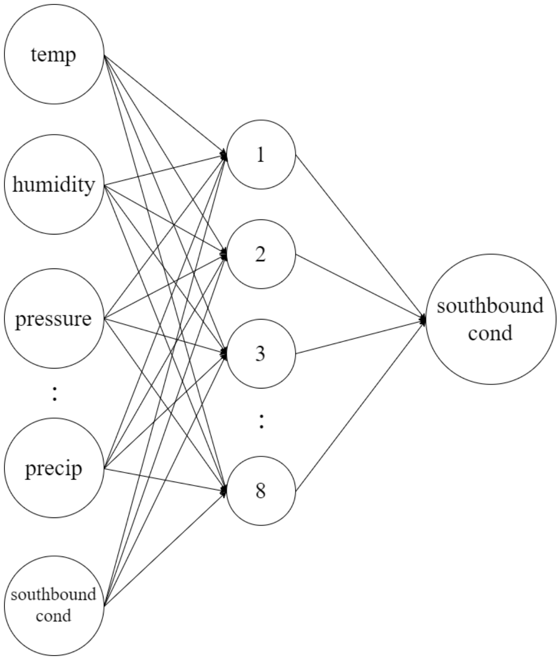
D.3.4 Training the Model

In preparation for training the model, we configured the model's optimizer, loss function, and evaluation metrics. For its optimizer, we set it to use the Adam optimizer, having a learning rate of 0.001 and a decay factor of 0.0. Meanwhile, for its loss function, we used mean squared error (MSE). Finally, for its evaluation metrics, we used mean absolute percentage error (MAPE). Using the training dataset, for both experiments, we trained the model using 1000 epochs and a batch size of 5.

D.3.5 Evaluation of the Model

For the first experiment, where the model only uses weather variable features as its training features, the model generated a root-mean-square error (RMSE) of 6.57%

Figure D.4: ANN Model for Predicting Southbound Traffic with Feature Fusion



and a MAPE of 57,787,420%. On the other hand, the second experiment, where the model uses feature fusion for the southbound traffic condition and weather variables as its training features, generated an RMSE of 0.0001% and a MAPE of 8,396%. Although the MAPE is quite high due to the lack of epochs, we could observe how massive the difference is between the first model and the second model. Given that, we can conclude that using feature fusion can significantly increase the accuracy of the model.

References

Ahlgren, P., Jarneving, B., & Rousseau, R. (2003). Requirements for a cocitation similarity measure, with special reference to Pearson's correlation coefficient. *Journal of the American Society for Information Science and Technology*, *54*(6), 550–560.

Armstrong, J. S., & Collopy, F. (1992). Error measures for generalizing about forecasting methods: Empirical comparisons. *International journal of forecasting*, *8*(1), 69–80.

Bando, M., Hasebe, K., Nakayama, A., Shibata, A., & Sugiyama, Y. (1995). Dynamical model of traffic congestion and numerical simulation. *Physical Review E*, *51*(2), 1035–1042.

Biham, O., Middleton, A. A., & Levine, D. (1992). Self-organization and a dynamical transition in traffic-flow models. *Physical Review E*, *46*(10), 46–50.

Bossé, É., Valin, P., Boury-Brisset, A.-C., & Grenier, D. (2006). Exploitation of a priori knowledge for information fusion. *Information Fusion*, *7*(2), 161–175.

Bovy, P. H., Salomon, I., et al. (2002). Congestion in europe: Measurements, patterns and policies. *Chapters*.

Castanedo, F. (2013). A review of data fusion techniques. *The Scientific World Journal*, *2013*.

Chai, T., & Daxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature. *Geoscientific Model Development*, *7*(3), 1247–1250.

Climate of the philippines. (n.d.). Retrieved from `https://kidlat.pagasa.dost.gov.ph/index.php/climate-of-the-philippines`

Constantinidis, A., Fairhurst, M. C., & Rahman, A. F. R. (2001). A new multi-expert decision combination algorithm and its application to the detection of circumscribed masses in digital mammograms. *Pattern Recognition*, *34*(8), 1527–1537.

Dai, X., & Khorram, S. (1999). Data fusion using artificial neural networks: a case study on multitemporal change analysis. *Computers, Environment and Urban Systems*, *23*(1), 19–31.

Dasarathy, B. V. (1997). Sensor fusion potential exploitation-innovative architectures and illustrative applications. *Proceedings of the IEEE*, *85*(1), 24–38. doi: 10.1109/5.554206

D'Mello, S. K., & Graesser, A. (2010, Apr). Multimodal semi-automated affect detection from conversational cues, gross body language, and facial features. *User Modeling and User-Adapted Interaction*, *20*(2), 147–187. doi: 10.1007/s11257-010-9074-4

Downs, A. (2005). *Still stuck in traffic: coping with peak-hour traffic congestion*. Brookings Institution Press.

Dunne, S., & Ghosh, B. (2013, March). Weather adaptive traffic prediction using neurowavelet models. *IEEE Transactions on Intelligent Transportation Systems*, *14*(1), 370-379. doi: 10.1109/TITS.2012.2225049

Fincher, D. W., & Mix, D. F. (1990). Multi-sensor data fusion using neural networks. In *Systems, man and cybernetics, 1990. conference proceedings., ieee international conference on* (pp. 835–838).

Fischer, A., & Igel, C. (2014). Training restricted Boltzmann machines: An introduction. *Pattern Recognition*. doi: 10.1016/j.patcog.2013.05.025

Gauthier, T. D. (2001). Detecting trends using spearman's rank correlation coefficient. *Environmental Forensics*, *2*(4), 359–362.

Geetha, K., & Radhakrishnan, V. (2013, May). Multimodal biometric system: A feature level fusion approach. *International Journal of Computer Applications*, *71*(4), 25–29. doi: 10.5120/12347-8635

Gu, Y., Qian, Z. S., & Chen, F. (2016). From twitter to detector: Real-time traffic incident detection using social media data. *Transportation research part C: emerging technologies*, *67*, 321–342.

Hall, D., & Llinas, J. (2001). *Multisensor data fusion*. CRC press.

Hamad, K., Ali Khalil, M., & Shanableh, A. (2017). Modeling roadway traffic noise in a hot climate using artificial neural networks. *Transportation Research Part D: Transport and Environment*. doi: 10.1016/j.trd.2017.04.014

Hauke, J., & Kossowski, T. (2011). Comparison of Values of Pearson's and Spearman's Correlation Coefficients on the Same Sets of Data. *Quaestiones Geographicae*, *30*(2). Retrieved from `http://www.degruyter.com/view/j/quageo.2011.30.issue-2/v10117-011-0021-1/v10117-011-0021-1.xml` doi: 10.2478/v10117-011-0021-1

Hu, W., Yan, L., Wang, H., Du, B., & Tao, D. (2016). Real-time traffic jams prediction inspired by Biham, Middleton and Levine (BML) model. *Information Sciences*, *381*, 209–228.

Hueper, J., Dervisoglu, G., Muralidharan, A., Gomes, G., Horowitz, R., & Varaiya, P. (2009). Macroscopic modeling and simulation of freeway traffic flow. *IFAC Proceedings Volumes*, *42*(15), 112 - 116. Retrieved from `http://www.sciencedirect.com/science/article/pii/S1474667016317839` (12th IFAC Symposium on Control in Transportation

Systems) doi: https://doi.org/10.3182/20090902-3-US-2007.0078

Hunter, A., Kennedy, L., Henry, J., & Ferguson, I. (2000). Application of neural networks and sensitivity analysis to improved prediction of trauma survival. *Computer Methods and Programs in Biomedicine*, *62*(1), 11–19. doi: 10.1016/s0169-2607(99)00046-2

Jain, V., Sharma, A., & Subramanian, L. (2012). Road traffic congestion in the developing world. In *Proceedings of the 2nd acm symposium on computing for development* (pp. 11:1–11:10). New York, NY, USA: ACM. Retrieved from `http://doi.acm.org/10.1145/2160601.2160616` doi: 10.1145/2160601.2160616

Jimenez, L. O., Morales-Morell, A., & Creus, A. (1999). Classification of hyperdimensional data based on feature and decision fusion approaches using projection pursuit, majority voting, and neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, *37*(3), 1360–1366.

King, R. C., Villeneuve, E., White, R. J., Sherratt, R. S., Holderbaum, W., & Harwin, W. S. (2017). Application of data fusion techniques and technologies for wearable health monitoring. *Medical Engineering & Physics*, *42*, 1–12. doi: 10.1016/j.medengphy.2016.12.011

Koesdwiady, A., Soua, R., & Karray, F. (2016, Dec). Improving traffic flow prediction with weather information in connected cars: A deep learning approach. *IEEE Transactions on Vehicular Technology*, *65*(12), 9508-9517. doi: 10.1109/TVT.2016.2585575

Lee, J., Hong, B., Kyungmin, L., & Yang-Ja, J. (2015). A prediction mode of traffic congestion using weather data. *IEEE Computer Society*, 81–88.

Lee, W.-H., Tseng, S.-S., & Tsai, S.-H. (2009). A knowledge based real-time travel time prediction system for urban network. *Expert Systems with Applications*, *36*(3), 4239 - 4247. Retrieved from `http://www.sciencedirect.com/science/article/pii/S0957417408001875` doi: http://dx.doi.org/10.1016/j.eswa.2008.03.018

Liu, D., Cho, S. Y., Sun, D. M., & Qiu, Z. D. (2010). A Spearman correlation coefficient ranking for matching-score fusion on speaker recognition. *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, 736–741.

Liu, J., Zhou, C., & Chen, J. (2008). Wdcm: a workday calendar model for workflows in service grid environments. *Concurrency and Computation: Practice and Experience*, *20*(4), 377–392.

Liu, Z., Li, Z., Li, M., Xing, W., & Lu, D. (2016). Mining road network correlation for traffic estimation via compressive sensing. *IEEE Transactions on Intelligent Transportation Systems*, *17*(7), 1880–1893.

Mahmud, K., Gope1, K., & Chowdhury, S. M. R. (2012). Possible causes & solutions of traffic jam and their impact on the economy of Dhaka city. *Journal of Management and Sustainability*, *2*(2), 112–135. doi: 10.5539/

jms.v2n2p112

Mahmud, K., & Town, G. E. (2016). A review of computer tools for modeling electric vehicle energy requirements and their impact on power distribution networks. *Appliied Energy*, *172*, 337–359.

Mangai, U. G., Samanta, S., Das, S., & Chowdhury, P. R. (2010). A survey of decision fusion and feature fusion strategies for pattern classification. *IETE Technical review*, *27*(4), 293–307.

May, A. D. (1990). *Traffic flow fundamentals.*

Meurant, G., & Meurant, G. (1992). Multisensor fusion. In *Data fusion in robotics & machine intelligence* (p. 41). Academic Press.

More, R., Mugal, A., Rajgure, S., Adhao, R. B., & Pachghare, V. K. (2016). Road traffic prediction and congestion control using artificial neural networks. In *2016 international conference on computing, analytics and security trends (cast)* (pp. 52–57).

Pan, B., Demiryurek, U., & Shahabi, C. (2012). Utilizing real-world transportation data for accurate traffic prediction. In *Data mining (icdm), 2012 ieee 12th international conference on* (pp. 595–604).

Refenes, A. N., Zapranis, A., & Francis, G. (1994). Stock performance modeling using neural networks: a comparative study with regression models. *Neural networks*, *7*(2), 375–388.

Regidor, J. R. F. (2013). Traffic congestion in metro manila: Is the uvvrp still effective? *Philippine Engineering Journal*, *34*(1), 66–75.

Saltelli, A., Chan, K., & Scott, E. M. (2000). *Sensitivity analysis.* New York: Wiley.

Saputri, T. R. D., & Lee, S.-W. (2013). Using artificial neural networks for predicting traffic conditions: A learning algorithm fro long-term time series forecasting. *Journal of Convergence Information Technology (JCIT)*, *8*(14).

Sohn, S. Y., & Lee, S. H. (2003). Data fusion, ensemble and clustering to improve the classification accuracy for the severity of road traffic accidents in korea. *Safety Science*, *41*(1), 1–14.

Sommer, M., Tomforde, S., & Hahner, J. (2013). Using a neural network for forecasting in an organic traffic control management system. In *ESOS.*

Tse, W. L. (2003). Knowledge-based algorithm for daily thermal load prediction of a building. *Energy Engineering*, *100*(5), 6-28. Retrieved from `http://www.tandfonline.com/doi/abs/10.1080/01998590309509243` doi: 10.1080/01998590309509243

Varangis, P., Skees, J., & Barnett, B. (2003). Weather indexes for developing countries. *Forest*, *4*, 3–754.

Vuchic, V. R., Bruun, E. C., Krstanoski, N. B., Shin, Y. E., Kikuchi, S., Chakroborty, P., & Perincherry, V. (1994). The bus transit system: Its underutilized potential.

Wang, C., Quddus, M. A., & Ison, S. G. (2009). Impact of traffic congestion on road accidents: A spatial analysis of the m25 motorway in england. *Accident Analysis & Prevention*, *41*(4), 798–808.

Wang, C., Quddus, M. A., & Ison, S. G. (2013). The effect of traffic and road characteristics on road safety: A review and future research direction. *Safety Science*, *57*, 264–275.

Yan, Z.-Z., Yan, X.-P., Xie, L., & Wang, Z. (2011). The research of weighted-average fusion method in inland traffic flow detection. *Information Computing and Applications Lecture Notes in Computer Science*, 89–96. doi: 10.1007/978-3-642-25255-6_12

Zar, J. (1972). Significance Testing of the Spearman Rank Correlation Coefficient. *Journal of the American Statistical Association*, *67*(339), 578–580.

Zhang, N., Ding, S., Zhang, J., & Xue, Y. (2017). An overview on restricted boltzmann machines. *Neurocomputing*. Retrieved from `http://www.sciencedirect.com/science/article/pii/S0925231217315849` doi: https://doi.org/10.1016/j.neucom.2017.09.065

Zhao, L., Lai, Y.-C., Park, K., & Ye, N. (2005). Onset of traffic congestion in complex networkson. *Physical Review E*, *71*(2), 026125.