Model-Free Bridge-Based Vehicle Classification

by

Grant Rutherford

A Thesis submitted to the Faculty of Graduate Studies of

The University of Manitoba

in partial fulfilment of the requirements of the degree of

MASTER OF SCIENCE

Department of Electrical and Computer Engineering

University of Manitoba

Winnipeg

Model-Free Bridge-Based Vehicle Classification

BY

Grant Rutherford

A Thesis/Practicum submitted to the Faculty of Graduate Studies of The University of Manitoba in partial fulfillment of the requirement of the degree

Of

MASTER OF SCIENCE

Grant Rutherford © 2008

Abstract

This thesis investigates the use of a sensor network on a bridge span to determine vehicle parameters such as weight and classification. An algorithm is developed and used to identify vehicle events on the bridge. These events are then matched with manual classification labels in order to study the features that distinguish various vehicle classes.

A number of features are extracted from both labelled and unlabeled event files. General trends in the features over time and temperature are discussed, and an attempt is made to estimate the weight of the vehicle from simple features.

Finally, perceptron and radial basis function neural networks are used to classify vehicle events based on the extracted features. A search of the parameter and feature space is performed in order to find the best network options. Trials with training and test data from the same and different seasons are compared. In addition, the vehicle weight estimate is improved through the use of the classification networks.

Model-Free Bridge-Based Vehicle Classification

Grant Rutherford

August 11, 2008

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem

An interesting development in the field of civil engineering is the practice of Structural Health Monitoring. As infrastructure ages, and as new design methods and materials are being developed, designers of large infrastructure projects are increasingly using strain sensors to monitor the health of their structures. These sensors can be used to verify the changing condition of a structure over time.

Since structures are now being built with sensors already incorporated into them, it is useful to consider what other information may be extracted from the data that they will collect. One interesting application is the problem of vehicle classification and weighing on bridges.

One significant factor in the health of a bridge is the number of large, heavy vehicles that pass over it [8]. Traffic engineers are often quite interested in understanding what type and weight of vehicles travel over particular structures. To this end, many commercial weigh-in-motion (WIM) and vehicle classification systems exist [1].

One type of WIM system currently being investigated involves the use of electrical strain sensors embedded in the longitudinal spans of a bridge to estimate vehicle parameters. In fact, commercial systems using this technology have already been patented [2]. As this technology improves and these systems become more commonplace, it will be important to have a sound understanding of the challenges and limitations inherent in their design.

To this end, this thesis will focus on a study of the stability and reliability of various parameters which would possibly be used in a practical bridge-based WIM or bridge-based vehicle classifier. Particular attention will be paid to changes in the parameters due to temperature and aging of the structure and sensors.

In order to do this, the problem of vehicle classification was investigated. Sensor data was collected from an instrumented span of the North Perimeter Bridge over the Red River in Winnipeg, Manitoba. This data was then correlated with manual vehicle classifications from an observer on site. All of this data collection was performed in both the summer and winter seasons in order to investigate the effects of temperature and time on classification. A number of parameters were chosen based on their ability to predict the class of vehicle which travelled over the bridge, and a variety of neural networks were trained with the data to make class predictions based on new data.

Investigations of the classification errors using training and test data from the same season and from different seasons were compared in order to determine the effects of changes to the parameters due to long term effects.

1.2 Prior Work

There are a number of previous studies on using strain gauges on bridges to collect information about the weight and type of vehicles that pass over the bridge. An article by Moses (1979) describes a WIM system using instrumented bridge spans to measure vehicle and axle weights using a least-squared algorithm [9]. In this application, tape switches were used to determine the speed and axle spacings of the vehicle, and strain measurements were used to estimate the weight.

Peil (2005) describe an analysis of vehicle weights by type. They divide their vehicle types into four classes: Cars, light or empty trucks, ordinary trucks, and heavy trucks. They distinguish between the four types by generating a histogram of vehicle weights and using discriminant analysis to estimate thresholds of 3, 17, and 28.5 metric tons [12]. The purpose of their analysis is to aid in the estimation of damage to structures by vehicle loading.

Loren Card (2004) uses a unsupervised neural network to identify interesting events on a number of instrumented structures. This system is flexible in that it doesn't require the input of a model of the structure, and has been shown to work with structures as diverse as bridges and statues. The described system uses input from multiple strain sensors and accelerometers to provide a novelty index of each interval of time [4]. Such a system could be used to identify vehicle crossings or overweight vehicles on bridges. This thesis uses a modified version of the concept of the novelty index in order to identify vehicles for classification.

A model of the static and dynamic loading of a bridge under various vehicle loads was done by Leming and Stalford (2003). They analyzed the theoretical response of the bridge under many conditions, and came to the conclusion that the truck weight could be determined quite well [7]. However, their results required an accurate model of the truck and bridge, and they did not test their results against measurements in the field.

Chapter 2

Data Collection

2.1 Bridge Sensor System

The data for this thesis was collected from the North Perimeter Highway Red River Bridge in Winnipeg, Manitoba, Canada. A section of the deck on this bridge was replaced and electrical strain gauges were added to the structure. The primary purpose of these sensors was to evaluate the quality of an experimental deck design. The new deck uses a GFRP-reinforced steel free design.

Electrical strain gauges (ESGs) were installed in a number of places on the deck, straps, and girders of the bridge section. Specifically, fifteen ESGs were installed on girders, with five near the abutment and ten in the middle of the span. Twenty ESGs were installed on straps between the girders around the mid-span, with five sensors per girder spacing. Additionally, fourteen sensors were installed into the deck itself. See Figure 2.1 for sensor locations.

A number of these sensors were found to be unusable for this thesis, for a number of reasons. A few sensors appeared to be electrically disconnected, and showed only a constant value for each sample. A number of other sensors were completely insensitive to events on the bridge, which could indicate problems with the sensor itself, the attachment, or the location of the sensor. After removing these sensors from the list, nine girder ESGs, thirteen strap ESGs, and five deck ESGs remained.

The instrumented deck span is 25m in length, and covers two lanes in the eastbound direction. There is also a merging lane which ends right before the span begins. There is a slight uphill grade, and the combination of this

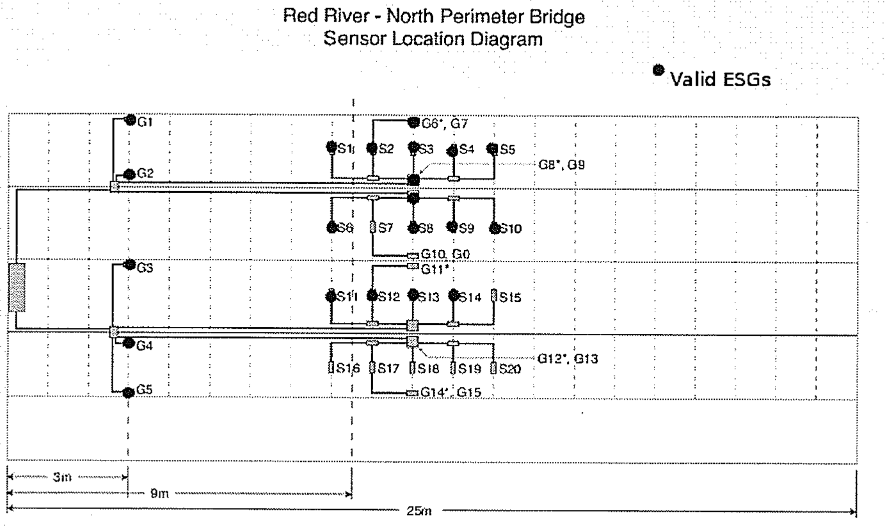

IMAGE CREDIT: D. MCNEILL

Figure 2.1: Sensor locations on north perimeter red river bridge

and the merging lane means that vehicles typically approach the span at a variety of speeds. Merging traffic is typically slower, while the rest of the traffic moves near the posted speed limit of 100 km/hr.

The ESGs were sampled continuously by a structural health monitoring system at a rate of 100 samples per second. This data is moved immediately over the Internet to a database at the University of Manitoba. Data for this thesis was downloaded from the offline copy of this database over a range of dates from July 2007 to February 2008.

2.2 Event Detection

The first challenge that this thesis attempts to solve is the problem of distinguishing vehicle events from normal loads. With twenty-seven sensors sending 100 samples per second, over 900 Mb of data is generated by the structure every day (after discarding useless sensors). Obviously, processing this much data would be problematic, so it is necessary to separate heavy vehicle events from the background traffic.

2.2.1 Event Detection Algorithm

Prior work in this area has identified that "normal" output for this type of structure is a relatively flat signal [4]. With this in mind, a novelty detection algorithm was developed that attempts to distinguish how much energy is in the signal relative to normal deviations.

This algorithm is based on a number of running averages, implemented using digital low pass IIR filters. The formula for updating these averages is:

$$y_t = y_{t-1} + \frac{x_t - y_{t-1}}{\tau} \tag{2.1}$$

Where y_t is the new "average" value, y_{t-1} is the most recent average value, x_t is the most recent input, and τ is the time constant of the average. Using this formula, y_t will track gradual changes in x_t.

One downside to this algorithm is that it needs to be given an initial value when the system is started or restarted. In order to minimize the impact of restarts on the system, the time constant τ is initially set to be 1, and is gradually increased with each new sample until the desired time constant is reached. This means that a "normal" average of the first points are used until a number of samples equal to the time constant have been collected.

The first average collected for each of the 27 sensors is the average value of the signal, $\mu_{t,k}$. The subscript k in this value indicates that the average is being collected for the kth sensor. This average is collected because strain is measured relative to a base strain. That is, the strain induced by an event is measured relative to the strain that existed before the event. In order to eliminate gradual, temperature-related strain drift from the measurements, this average is subtracted from all further novelty calculations. The time constant of this average is set to 2000 samples, or 20 seconds.

$$\mu_{t,k} = \mu_{t-1,k} + \frac{x_{t,k} - \mu_{t-1,k}}{2000} \tag{2.2}$$

The variance of the signal from this average, $s_{t,k}$, is then calculated as an estimate of the instantaneous energy of the signal:

$$s_{t,k} = (x_{t,k} - \mu_{t,k})^2 \tag{2.3}$$

Two running averages of this variance are then collected. One of these, $s_{t,k}^{long}$, uses a long time constant, and is meant to estimate the normal energy of the signal. In this way, sensors that are normally more sensitive can be

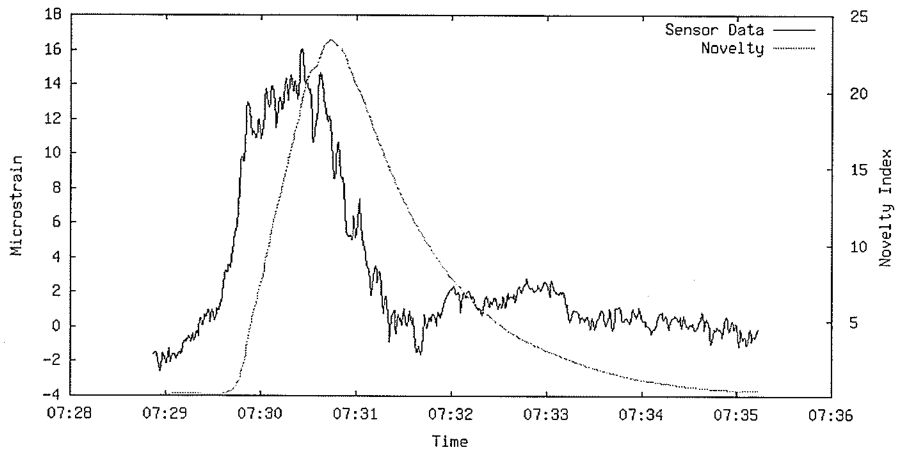

Figure 2.2: Typical novelty index for vehicle event

given a lower significance, while sensors that usually have very little variance can be given more weight. The time constant of this running average is 30000 samples, or 5 minutes.

$$s_{t,k}^{long} = s_{t-1,k}^{long} + \frac{s_{t,k} - s_{t-1,k}^{long}}{30000} \tag{2.4}$$

The second running average of variance, $s_{t,k}^{short}$, is intended to smooth out the energy of the signal over the duration of an event, so that the beginning and end of the event can be more easily established. A time constant of 100 samples, or 1 second, is used for this measurement.

$$s_{t,k}^{short} = s_{t-1,k}^{short} + \frac{s_{t,k} - s_{t-1,k}^{short}}{100} \tag{2.5}$$

The instantaneous novelty index $n_{t,k}$ is then calculated for each sensor by taking the ratio of the short and long variance averages. An example of a typical novelty index is given in Figure 2.2.

$$n_{t,k} = \frac{s_{t,k}^{short}}{s_{t,k}^{long}} \tag{2.6}$$

Finally, the average novelty index of all 27 sensors is computed to generate a instantaneous global novelty index for the entire structure:

$$N_t = \frac{\sum_{k=1}^{27} n_{t,k}}{27} \tag{2.7}$$

2.2.2 Event Novelty Thresholds

Once the instantaneous novelty index is found for each sample, it can be used to identify interesting events. However, it is important to select a reasonable value for the cut-off threshold. If this threshold is set too low, then too many events will be detected which do not actually correspond to heavy vehicles. Also, events which occur in quick succession are more likely to be lumped into a single multiple vehicle event. Conversely, if the threshold is set too high then events may be missed.

Also, in order to avoid very short events, hysteresis is necessary. Trial and error identified that using a threshold of 1.4 to identify the beginning of the event and 1.3 to identify the end of the event produced a reasonable trade-off between very few undetected vehicles and not too many detections of non-events.

The data from each sensor for the event, along with one second of data before and after the event, was separated into a file for further processing. The running average value of the signal at the beginning of the event was subtracted from the data in order to give it a zero value in approximately the same region as the unexcited sensor signal.

2.2.3 Novelty Algorithm Problems

A few problems were noticed when using the novelty detection algorithm on large amounts of real-life data. In particular, the data from the system included several missing sections of time and regular jumps in the values from the sensors.

Missing values were handled by simply ignoring events which spanned the time jump, and resetting the averages when the data returned. This "resetting" was handled as described in section 2.2.1.

Jumps in the values of the sensors occurred in two ways. First of all, at times all of the sensors in the system would change value dramatically over a couple of samples. In order to avoid missing a large number of events following such a change, an extremely short running average with a time constant of only 50 samples (1/2 second) was kept for each sensor, and any deviations of more than 1000 microstrain from this running average triggered a reset of all of the running averages used in the algorithm. A few sensors which were identified as problematic were excluded from this algorithm.

In addition to these global jumps, occasionally a single sensor would jump momentarily to an extremely large value. This was probably due to a loose connection in the wiring somewhere. Since these events were more common, it was decided that a global reset would not be necessary. Since the novelty index of the system returns to "normal" with an exponential decay, extremely large jumps in any sensor produced extremely long events (since the novelty index would take a long time to decay below 1.3). Therefore, in order to remove these events from the data, all events that had a duration of more than 10 seconds were discarded.

2.3 Manual Event Classification

In addition to automatic detection of vehicle events, a number of vehicle events were observed and categorized manually. This occurred on four separate dates.

2.3.1 July Data

On July 12th, 2007, a pair of vehicles of known weights were sent over the bridge in a series of controlled tests. This data was being gathered for a separate research goal, but it will also be used by this thesis in order to verify the accuracy of weight estimates.

The smaller vehicle in these tests (Figure 2.3) consisted of a single trailer and had a total weight of 395.6 kN, or 40340 kg. The larger vehicle (Figure 2.4) consisted of two trailers and had a total weight of 625.5 kN, or 63783 kg. These vehicles were sent over the bridge with a variety of speeds and lane configurations, and the strain results were collected. A total of 11 single vehicle events were captured, as well as 4 multiple vehicle events.

2.3.2 August Data

The first major data collecting day occurred on August 22nd, 2007. The purpose of manual event classification was to provide labelled data points so that generalizations could be made about certain classes of events. For the

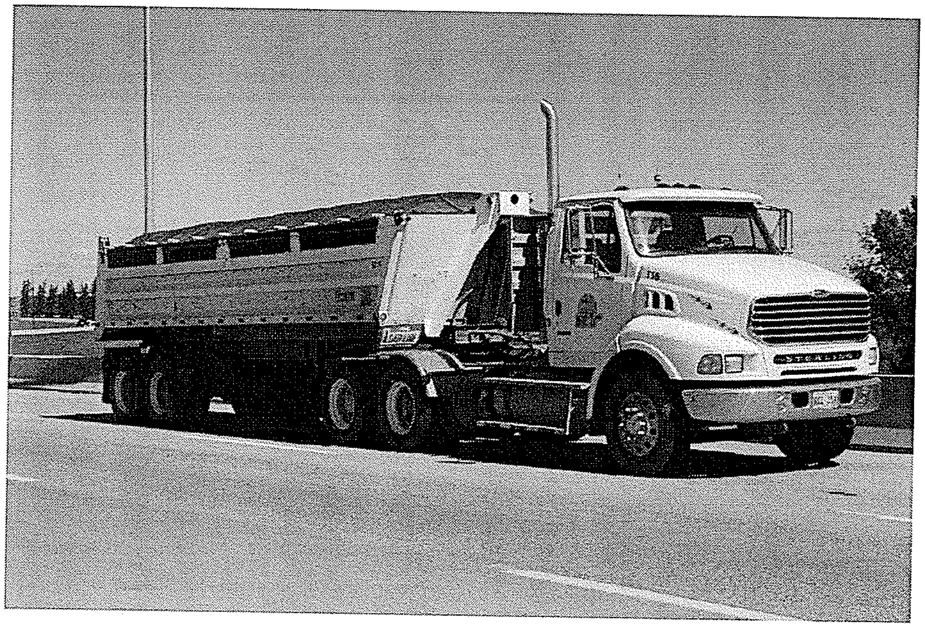

IMAGE CREDIT: D. MCNEILL

Figure 2.3: Smaller test vehicle

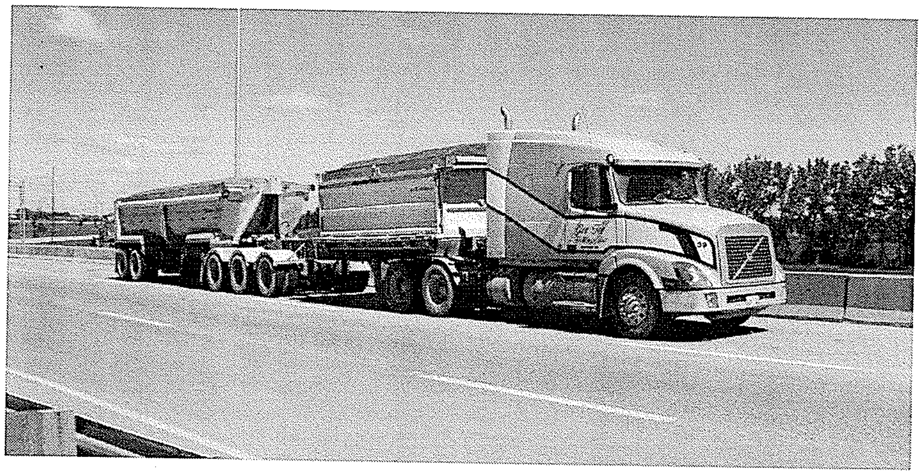

IMAGE CREDIT: D. MCNEILL

Figure 2.4: Larger test vehicle

purposes of this thesis, six classes of events were observed: Normal and large trucks, and other large vehicles, each of which were further divided into the left and right lanes.

Normal trucks were defined to be semi-trailer trucks with a single trailer only. This class was further subdivided into normal trucks in the left (passing) lane and normal trucks in the right lane. Please note that left and right lanes are relative to the traffic's point of view. This class included both loaded and unloaded semi-trailers, and all varieties of semi-trailer, including box, tanker, and flatbed trailers. This class vaguely conforms to FHWA Classifications 8, 9, and 10 [11]. The vehicle in Figure 2.3 would be an example of this class.

Large trucks were defined to be semi-trailer trucks with two or more trailers. Once again, this class was divided into left and right lane events. Also, a variety of trailer types were observed in this class, the most common being tanker vehicles. This class vaguely conforms to FHWA Classifications 11, 12, and 13 [11]. The vehicle in Figure 2.4 would be an example of this class.

The final class of vehicles observed were all other large vehicles which appeared as if they would trigger the novelty detection algorithm. This class included a huge variety of vehicles, including passenger trucks with a trailer, trucks with a non-detachable cargo space, cement trucks, and road-cleaning vehicles. Noteable exceptions which were not recorded in any class include passenger cars and trucks, vans, and ordinary semi-trailer trucks without a trailer.

A total of 604 events were observed in this data collection period and were also detected by the novelty detector. When the observation data was lined up with the data from the bridge sensors and the novelty detector, 168 of these vehicles were found to occur in the same event as another vehicle, and were therefore classified into 78 "multiple vehicle" events. In addition, the novelty detector identified 68 events which were not recorded by the manual observations but which occurred during the time period that the observer was present.

Of the 436 events that were properly labelled and were not detected as multiple vehicle events, 35 were in the large truck class, with 15 in the left lane and 20 in the right lane. Of the 288 vehicles in the normal truck class, 109 were in the left lane and 179 were in the right lane. Finally, 113 other large vehicles were observed, with 28 in the left lane and the remaining 85 in the right lane.

Correlating results from the manual observations with the novelty detector events was a non-trivial task. Because the clocks of the two systems were not perfectly aligned, software was written to find the peak correlation between the two inputs, and the manual observations were shifted by a few seconds so that they lined up better with the detected events. An automatic algorithm then went through all of the data and labelled the events appropriately. Approximately 10% of the manual observations had to be adjusted by up to 5 seconds afterwards in order to match detected events. This was necessary due to operator delays inherent in the manual observations.

2.3.3 February Data

The final data collecting days occurred on February 5th and 6th, 2008. Two days were necessary in order to collect the same amount of data due to the cold weather. The same classes and process were used as for the August data. Also, the same algorithms were used to line up the detected results with the observed vehicles.

A total of 564 events were observed in February which were also detected by the novelty detector. When the observation data was aligned with the data from the novelty detector, 142 of these vehicles were found to occur in 62 "multiple vehicle" events. The novelty detector also identified 176 events which were not observed manually.

Of the 422 events that were labelled and were not part of multiple vehicle events, 39 were in the large truck class, with 11 in the left lane and 28 in the right lane. The normal truck class had 232 vehicles, with 78 in the left lane and 154 in the right lane. Finally, 151 other large vehicles were observed, with 33 in the left lane and the remaining 118 in the right lane.

2.4 Filtering

For the purposes of extracting the load parameters from the signal, it is first necessary to eliminate extraneous factors such as high-frequency noise and the natural oscillations of the bridge and load.

Numerous parts of the system that is being observed by the ESG measurements tend to exhibit natural frequencies, and will vibrate at these frequencies. For example, the bridge span is supported between two supports and when excited by a heavy load it will tend to vibrate at a fairly low

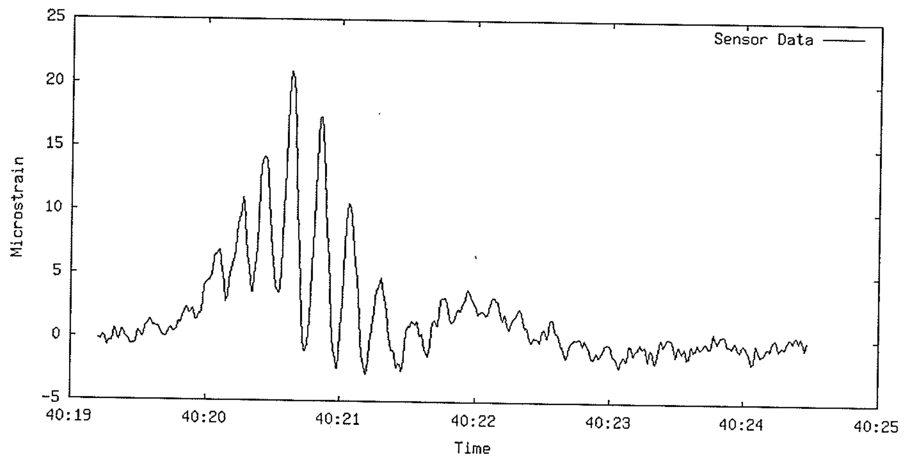

Figure 2.5: A typical example of a highly oscillatory event

frequency. Additionally, the suspension systems of vehicles will have their own natural frequency and the load applied to the bridge span will tend to oscillate as this suspension system vibrates [5].

While observing the outputs of the sensors during events, almost all events appeared to exhibit some natural frequency. A small subset of these appeared to have an extremely powerful vibration at some natural frequency (Figure 2.5).

Due to the fact that the frequency of these vibrations would depend on many factors, including temperature and bridge load, as well as vehicle mass and suspension system qualities, it was determined that filtering the data would be best performed on a case-by-case basis.

An algorithm was developed to determine the natural frequency of the event, if it exists. This was done in order to set the cutoff frequency of a digital filter to an appropriate level in order to filter this frequency out. The algorithm first took a series of FFT measurements using 100 sample Hamming windows [13]. Taking one such FFT every 10 samples (90% overlap), the peak of the FFT between 7.782 Hz and 38.911 Hz is found for each FFT. The average of these peak frequencies is then divided by two, and 1 Hz is subtracted. If this value is less than 6 Hz, then 6 Hz is used instead.

A low pass digital filter of order 512 is then created using another Hamming window. This filter is then applied to the signal. The values used by this algorithm were chosen by trial and error to do a good job of eliminating

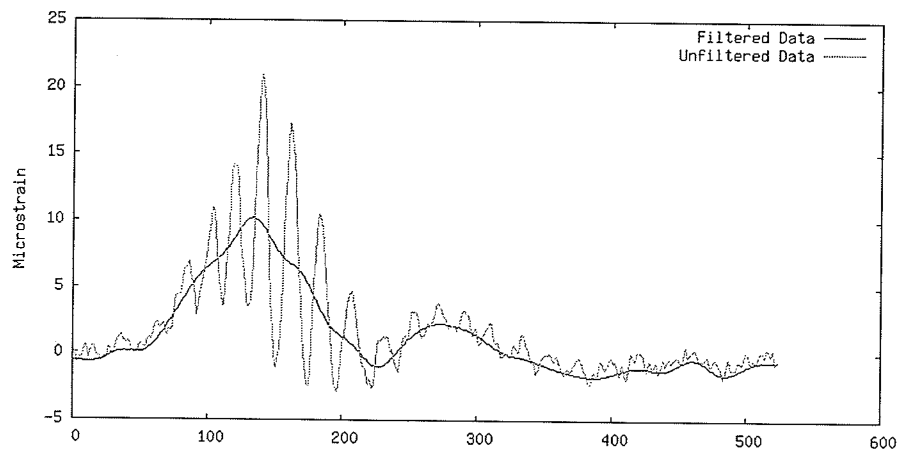

Figure 2.6: A noisy event and the effects of filtering

the noise without destroying the underlying load trends. An example of the results of the filter on the noisy signal from Figure 2.5 is given in Figure 2.6.

As a final step, the median value of the filtered data is subtracted from the entire signal. It was noticed that some signals had a non-zero base value, probably due to drift in the strain gauge that wasn't properly compensated for by the running averages described in section 2.2.1. This ensures that the zero-value is the true base value in the event. Note that both filtered and unfiltered data are used in the feature extraction process.

2.5 Feature Extraction

Once the data has been collected, sorted into events, labelled, and filtered, the next step is to extract features from the signal. Since data analysis by automated means such as neural networks would be extremely difficult with raw signal inputs of hundreds of samples from multiple sensors, it is beneficial to reduce the information presented to the system. In order to do this, features are extracted from the signal which are expected to contain the majority of the "interesting" information of the signal, and the rest of the information is discarded [3].

A number of different features were gathered from the data for this thesis. These include peak values, signal variance, cross-correlation values, various measures of the event length, an attempt to count the axle groups in the event, and the temperature.

2.5.1 Peak Value and Signal Variance

The two simplest features extracted were simply the peak value and the signal variance. Both of these parameters were extracted from the unfiltered data, and were intended to give an estimate of the magnitude of the vehicle load.

These parameters were gathered for all nine of the valid girder ESG sensors, as well as one of the deck ESG sensors.

In addition, two aggregate values were calculated using these parameters. These values were intended to give a good measurement of which lane the vehicle was travelling in. The ratio of the sum of girder ESGs 3, 4, and 5 on the right hand side of the bridge, divided by the sum of girder ESGs 1 and 7 on the left hand side of the bridge was calculated for both the peak value and the variance. Please see the sensor diagram in Figure 2.1 for the exact locations of these sensors.

The peak of the filtered data was also calculated, but it was not used as a feature. Instead, it was used to identify the start and end time of the actual event within the event data. A threshold of 10% of the peak value was used, and the final time that this threshold is crossed before the peak is identified as the start of the event. The first time that the threshold is crossed after the peak is identified as the end of the event.

2.5.2 Cross Correlation

The cross correlation between sensor signals from ESGs at the abutment and ESGs in the middle of the span was calculated in an attempt to estimate the vehicle speed. The peak of the cross correlation of the first 200 samples (2 seconds) was taken between girder ESGs 2 and 8. These two sensors were chosen because they both react strongly to events in either lane. A restriction was added that this peak must lie in the first 50 samples of offset in order to reduce computational requirements and eliminate ridiculous results. This measurement was taken for both filtered and unfiltered data.

A related measurement, which will be referred to as the start time distance, involves calculating the difference in the start of the events from the 10% threshold of the filtered data. This measurement was also calculated between girder ESGs 2 and 8.

2.5.3 Event Length

A number of measurements of event length were attempted. The first and simplest measurement was simply the number of samples collected in the event by the novelty detector. While this is a measure of the length of the event data, it is effectively more of a measure of the event intensity, due to the exponential decay of the novelty index.

A fairly accurate measure of the actual length of the vehicle crossing can be obtained by finding the difference between the start and end times calculated around the peak of the filtered data. See section 2.5.1 for details. This measurement was performed for both girder ESGs 2 and 8.

The next event length measurement was performed on the unfiltered data from girder ESGs 2 and 8. It involved finding the peak of the sensor, and marking the point where the sensor value first exceeded 25% of the peak value, and the point where the sensor first dropped below 25% of the peak value. A higher threshold was used for the unfiltered event length measurement because there is more noise in the signal.

The event length and vehicle speed approximations were chosen because it is believed that the combination of these two parameters would give an automatic classifier an approximation of the vehicle length.

2.5.4 Bump Measurements

A final pair of features extracted from the filtered data were an attempt to identify and count axle groups. Since each axle group would add load as it rolled onto the bridge span, it is believed that there should be a distinguishable "bump" in the sensor data when this happens.

It was noticed that these bumps did appear in the data, but they did not always take the form of a local maximum. In some cases, as in Figure 2.7, you can clearly see that there are three axle groups, but only one of these is a local maximum. In order to identify the other two groups, the concavity of the graph must be investigated.

A numerical approximation of the concavity of a graph can be given by its numerical second derivative.

$$\frac{d^2 x_{t,k}}{dt^2} \approx x_{t+1,k} - 2x_{t,k} + x_{t-1,k} \tag{2.8}$$

When the value of this second derivative is above zero, the graph is con-

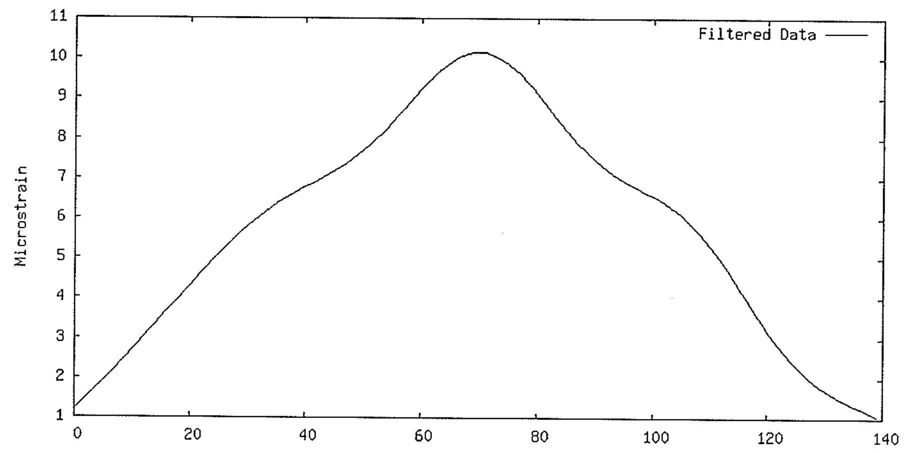

Figure 2.7: A filtered event with three axle groups

cave up. When it is below zero, the graph is concave down. The bumps in the signal are regions that are concave down, and have a local maximum absolute concavity. This means that the criteria for finding the peak of a bump is to look for a point where the second derivative is both less than zero, and at a local minimum.

In order to find the local minimum, the numerical derivative of the second derivative is found, giving the numerical third derivative. The positive zero crossings of this function, from negative values to positive values, will identify the local minima of the second derivative.

$$\frac{d^3 x_{t,k}}{dt^3} \approx x_{t+1,k} - 3x_{t,k} + 3x_{t-1,k} - x_{t-2,k} \tag{2.9}$$

As specified above, the criteria for identifying a time t where a bump has occurred will be the following:

$$\frac{d^3 x_{t,k}}{dt^3} \geq 0, \tag{2.10}$$

$$\frac{d^3 x_{t-1,k}}{dt^3} < 0, \tag{2.11}$$

$$\frac{d^2 x_{t,k}}{dt^2} < 0 \tag{2.12}$$

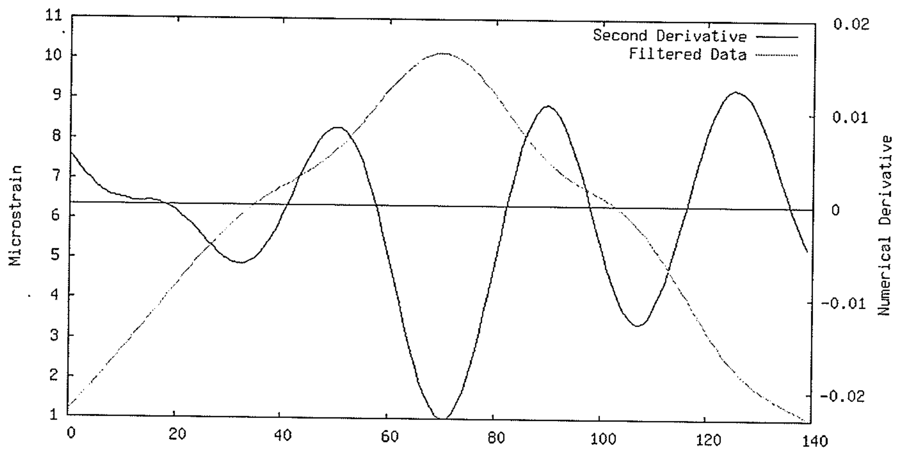

Figure 2.8: The second derivative is used to identify axle groups

Figure 2.8 illustrates the application of this algorithm to the signal in Figure 2.7. Four local minima of the second derivative are found, however the first of these is in the relatively straight section at the beginning of the graph. It is disqualified because the second derivative is positive at this point. So the algorithm counts three bumps at the correct locations on the graphs.

Two different features were gathered using this algorithm. The first of these simply counts the bumps that occur during the duration of an event. The second feature is intended to be a measurement of the vehicle speed and counts the time between the second and third bumps of the event. In both cases, only bumps that occur after the estimated start time of the event and before the estimated end time (from section 2.5.1) are used. Both of these features are collected for girder ESGs 2 and 8.

2.5.5 Temperature

A final feature attached to each event was the temperature. There were six temperature sensors on the bridge giving temperature readings once per second. The readings from all six sensors were averaged over each minute and collected on the server. Based on the time of each event, the appropriate temperature was also extracted as a feature.

Note that while this temperature reading may be less accurate than temperature data from official weather stations, it is more relevant because these temperature sensors are actually attached to the structure in the same areas as the strain sensors. Since the bridge likely has a very high thermal time constant, the temperature of the bridge will not necessarily be the same as the temperature of the air.

2.5.6 Feature Summary

Table 2.1 shows the 39 features used in the following sections. All of these features were extracted from each event, and stored in batch files for processing by statistical analysis and neural networks.

Table 2.1: Features extracted

ID	Feature	Sensor	Filtered
1	Event Timestamp	N/A	N/A
2	Event Class Label	N/A	N/A
3	Event Temperature	N/A	N/A
4	Novelty Detector Event Length	N/A	N/A
5	Filter Cutoff Frequency	Girder ESG 2	N/A
6	Filter Cutoff Frequency	Girder ESG 8	N/A
7	Peak measurement	Girder ESG 1	No
8	Peak measurement	Girder ESG 2	No
9	Peak measurement	Girder ESG 3	No
10	Peak measurement	Girder ESG 4	No
11	Peak measurement	Girder ESG 5	No
12	Peak measurement	Girder ESG 6	No
13	Peak measurement	Girder ESG 7	No
14	Peak measurement	Girder ESG 8	No
15	Peak measurement	Girder ESG 9	No
16	Peak measurement	Deck ESG	No
17	Signal Variance	Girder ESG 1	No
18	Signal Variance	Girder ESG 2	No
19	Signal Variance	Girder ESG 3	No
20	Signal Variance	Girder ESG 4	No
21	Signal Variance	Girder ESG 5	No
22	Signal Variance	Girder ESG 6	No
23	Signal Variance	Girder ESG 7	No
24	Signal Variance	Girder ESG 8	No
25	Signal Variance	Girder ESG 9	No
26	Signal Variance	Deck ESG	No
27	Ratio of Left Peaks to Right	Various	No
28	Ratio of Left Variances to Right	Various	No
29	Maximum Cross Correlation	Girder ESGs 2 and 8	No
30	Maximum Cross Correlation	Girder ESGs 2 and 8	Yes
31	Start Time Difference	Girder ESGs 2 and 8	Yes
32	Vehicle Length Estimate	Girder ESG 2	No
33	Vehicle Length Estimate	Girder ESG 8	No
34	Vehicle Length Estimate	Girder ESG 2	Yes
35	Vehicle Length Estimate	Girder ESG 8	Yes
36	Bump Count	Girder ESG 2	Yes
37	Bump Count	Girder ESG 8	Yes
38	Second to Third Bump Distance	Girder ESG 2	Yes
39	Second to Third Bump Distance	Girder ESG 8	Yes

Chapter 3

Data Analysis

3.1 Unlabeled Data

In addition to the labelled data collected on July 12th, August 22nd, and February 5th and 6th, at the time of this writing, the system has been collecting data continuously for over a year. In order to draw general conclusions about the bridge, traffic, and system, this data has been collected and analyzed for the period from August 23rd 2007 to February 4th 2008.

The raw data was downloaded from the database, and run through the novelty detection algorithm. A total of 379602 unique events were identified by the novelty detector during this period. Unfortunately, there are some periods of missing data in the results due to system errors. Specifically, a couple of days are missing in September, about a week is missing in November, and a few days at the beginning of January are missing. Due to the large volume of valid data, however, it is not expected that this missing data will be problematic.

In addition, 2 of these events were manually observed to be disrupting analysis due to extremely large microstrain values. Specifically, they recorded strain peaks of over 5000 microstrain on many sensors. It is believed that these are actually data collection glitches as described in section 2.2.3 that somehow got through the system. These two events were removed, leaving 379600 valid events.

Due to the large temperature variations that Winnipeg experiences, this data covers over 55 degrees of temperature readings as reported by the temperature sensors. This includes measurements at -25 degrees Celsius and 30 degrees Celsius. It covers over five months of aging, night and day cycles, and a large variety of traffic patterns. The rest of this chapter will be devoted to interesting results obtained from analyzing this data directly.

3.2 Event Frequency and Intensity

Interesting features of the data can be observed by plotting event data versus time. A plot of the number of events per hour versus time (Figure 3.1) confirms some expectations and raises some interesting questions. First of all, as expected, it can be seen that the number of events detected per hour is cyclic over the day, with more events during the day and less at night. Also, looking at the graph from August to February, it can be seen that the number of events detected stays relatively constant. If the assumption is made that there are no long term trends in the traffic size or frequency, this confirms that the bridge, sensors, and novelty detection algorithm are not becoming more or less sensitive over time.

Interestingly, though, there are some suprises. It was believed that the bridge would not be strongly effected by small passenger vehicles. However, on this graph there is a clear trend towards detecting more events during weekends and holidays, rather than less. Also, a clear peak can be seen on the dates of December 25th and 26th (Figure 3.3). It seems unlikely that shipping and other heavy traffic would be more prevalent during these days, so it seems possible that the system is also detecting smaller vehicles. The bridge does see fairly constant traffic of passenger vehicles, so obviously not every passenger vehicle is setting off the system. One possible conclusion is that some heavier passenger vehicles or vans set it off, or possibly a combination of many passenger vehicles on the span at once causes an event to be detected.

A look at average event intensity over time seems to confirm these conclusions. For the purpose of this analysis, event intensity is defined as the average of the peak measurements of all nine girder ESG sensors during the event. Figure 3.2 shows almost the opposite trends as Figure 3.1. While there is still a clear day to night cycle, average event intensity is clearly much higher on weekdays than on weekends. Also, the event intensity remains low for the entire week between Christmas and New Year's Day. In fact, on Christmas and Boxing Day, average event intensity stays the same during the day as it is at the middle of the night (see Figure 3.4 for details). This is despite the fact that those two days saw the highest number of events

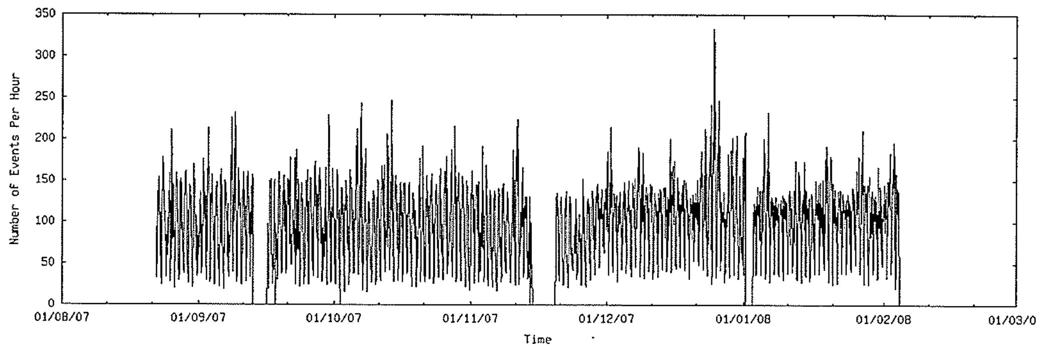

Figure 3.1: Number of events over entire test

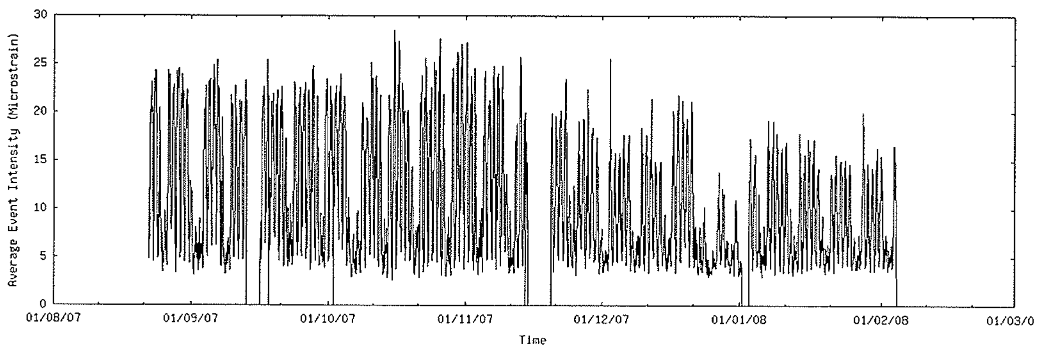

Figure 3.2: Event intensity over entire test

recorded per day over the entire data range.

Please note that when looking at Figures 3.3 and 3.4, the first day on the graph, December 1st 2007, was a Saturday. December 25th of that year was a Tuesday.

3.3 Event Temperature Trends

In addition to plotting event intensity and frequency versus time, trends in the data by temperature can also be analyzed. Figure 3.5 shows plots of average event intensity and number of events versus temperature.

Although this graph does show some trend towards larger events at higher temperatures, it doesn't seem likely that this is a result of the system becoming more sensitive at higher temperatures. Indeed, if that were the case then

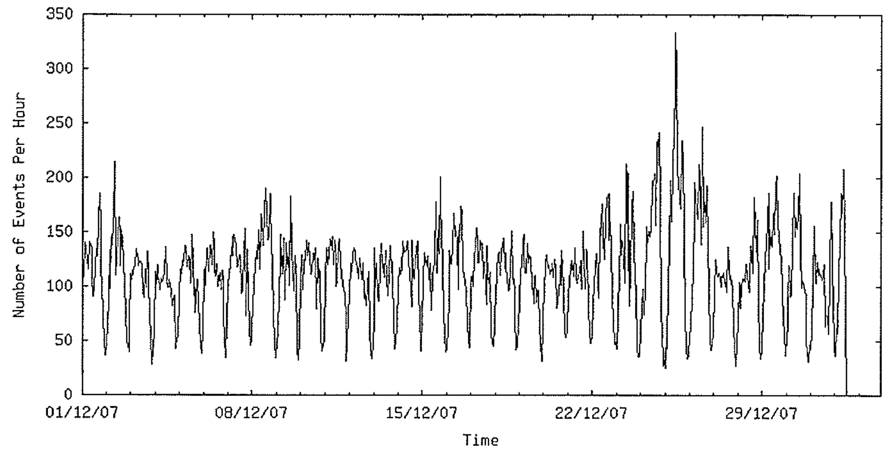

Figure 3.3: Number of events (December only)

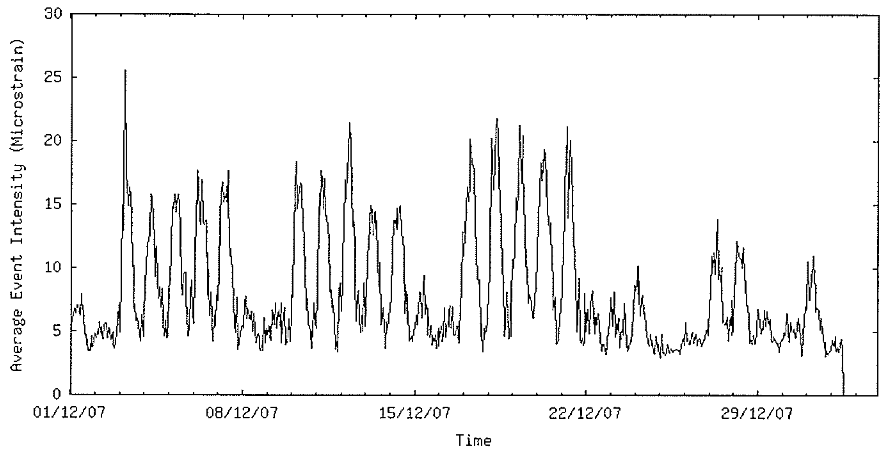

Figure 3.4: Event intensity (December only)

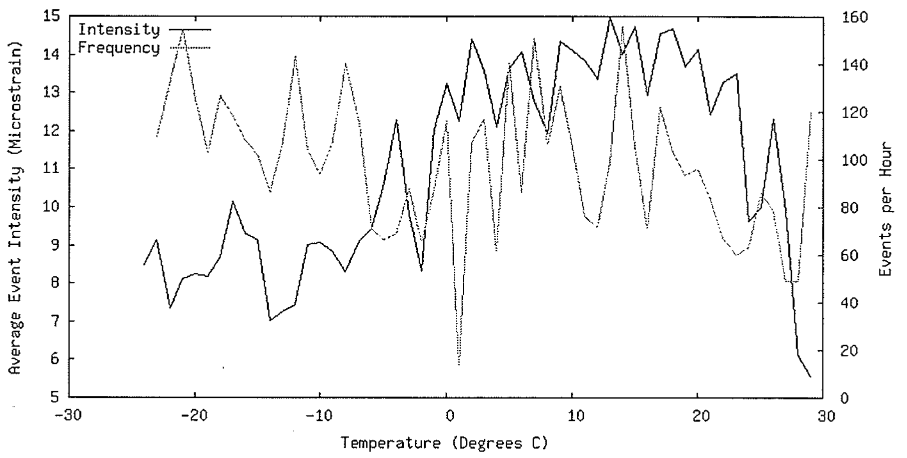

Figure 3.5: Temperature trend of event intensity

clearer trends in the time graphs would be expected in the previous section from August to February.

Perhaps the trend in this graph can simply be explained as an artifact of the daily periodicity of the traffic. Since the heaviest traffic occurs during the day, when temperatures are at their highest, it should not be suprising that when event intensity is graphed with respect to temperature, it appears that higher temperatures cause larger events. Rather, it should be concluded that higher temperatures and larger events simply tend to occur at the same times.

3.4 KDE Vehicle Weight Analysis

A final analysis of the unlabeled data is done with kernel density estimation [3]. This process is a means of smoothing out a number of discrete events in a parameter space in order to create an estimate of the probability function for an event occurring at any particular point in that parameter space. The probability function produced is a relative probability distribution only. It must be normalized in order to become a true probability distribution.

For the sake of simplicity, only a single parameter will be used. This reduces the process to its simplest form, with only one dimension. It also makes the results easy for a human to view and analyze. The parameter that will be estimated is the same event intensity parameter discussed in the

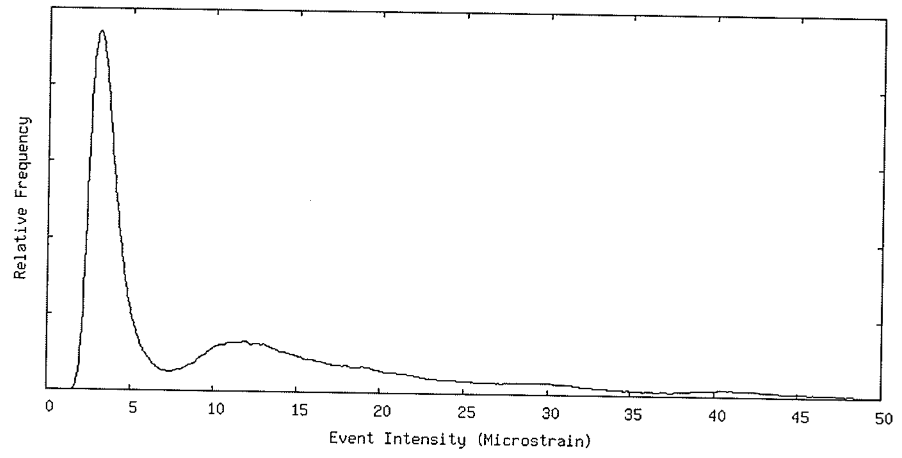

Figure 3.6: Kernel density estimation of event intensity probability function

previous sections, which is composed of the average of the peak measurements of the nine girder ESG sensors.

Figure 3.6 shows such a probability estimation using a gaussian kernel with a width parameter of 0.1 microstrain. Please note that due to outliers, the graph is not zero outside of the range presented, but it will appear to be zero at the resolution of the graph so the full range is not included.

Since it seems intuitive to assume that the peak sensor activation would be linearly related to the gross vehicle weight of the vehicle causing the event, it may be possible to use this graph to estimate a function for vehicle weight based on the event intensity. The maximum allowed weight for a vehicle in Manitoba is 62500 kg, and this bridge and the surrounding highways do not have any special weight restrictions [10]. If the small peak around 40 microstrain is assumed to represent multiple-vehicle events, and a margin is allowed to represent over-weight vehicles, a rough approximation can be made with the intensity of a 62500 kg vehicle at around 32 microstrain. Assuming a linear relationship, this translates to about 1953 kg per microstrain. Note that the choice of 32 microstrain was done manually due to a lack of a deterministic way to select this parameter.

In order to check the accuracy of this estimation, the controlled vehicle events observed during the July 12 data collection day (Table 3.1) can be used. Since these vehicles were of a known weight, it would be useful to use their strain intensity results to test this estimate. Looking at the re-

Table 3.1: Event intensity in microstrain from July vehicles

Trial	Lane	Speed	40340 kg Vehicle	63783 kg Vehicle
1	Right	25 km/h	32.308	29.443
2	Right	50 km/h	32.896	37.742
3	Right	75 km/h	33.020	39.937
4	Right	100 km/h	36.123	40.329
8	Left	100 km/h	44.668	54.569
9	Left	100 km/h	N/A	55.471
Average			35.803	42.915

sults in this table, it seems that the smaller vehicle implies a conversion of 1127 kg per microstrain. The larger vehicle implies a conversion of 1486 kg per microstrain. Also, during the July tests girder ESG sensor 6 was not functioning properly, so it was not included in the average.

Obviously, these results are not very consistent. It seems both that there is a large variation of strain readings for the same vehicle, as well as a non-linear relationship of some kind between average strain peaks and vehicle weight. Interestingly, it appears that the average peak strain reading is increasing during the trials. Perhaps there is some external factor influencing the readings over time. While this may cast some doubts on the usefulness of using strain peaks to estimate vehicle weight, there could simply be problems with the July data or a non-linear relationship between strain and weight.

Since Figure 3.6 includes every event that exceeded the novelty threshold on the novelty detector, it represents everything interesting that occurs on the bridge. This will include normal semi-trailer trucks, multiple vehicle events, sensor glitches, and quite likely many passenger vehicles (see section 3.2). It would also include vehicles in both the left and right lanes. In the next section a method of classifying these events will be investigated, and it may be possible to refine the probability estimation by class.

Chapter 4

Neural Network Analysis

The main goal of this thesis was to investigate the possibility of using data from the sensor network on the bridge to classify vehicles passing over the bridge. This section will discuss the use of neural networks to attempt this classification.

4.1 Classification Difficulties

First of all, it should be pointed out that this problem is not as simple as it might initially seem. While it does seem easy for a human operator to make classification decisions between such disparate classes as single and multi trailer trucks, and other vehicles, the sensor network does not have as clear a view of the situation as might be expected. As suggested by sections 2.4 and 3.4, the ability of the system to represent even an instantaneous loading of the bridge is questionable.

The features chosen in section 2.5 were selected with the goal of providing good separation between the vehicle classes. However, this problem was quickly discovered to be more difficult than it seems.

Figures 4.1, 4.2, 4.3, and 4.4 illustrate this problem. All four of these graphs were generated from the results of girder ESG 2 from the August data for events in the right lane. Although it seems reasonable to expect that single trailer trucks will be heavier than vehicles in the "Other" class, and lighter than multiple trailer trucks, these four graphs do not show this trend at all. The event in the "Other Vehicle" class has a peak microstrain of about 34, while the dual trailer truck has a peak microstrain of about

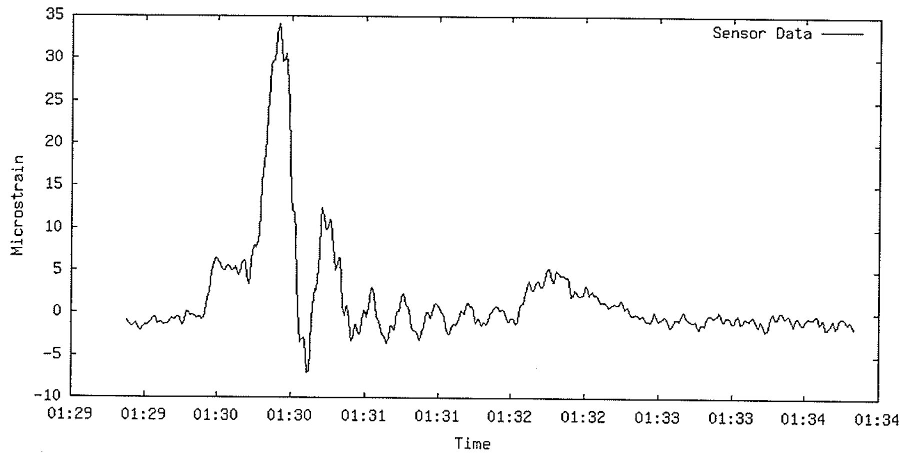

Figure 4.1: Example event in "Other Large Vehicle" class

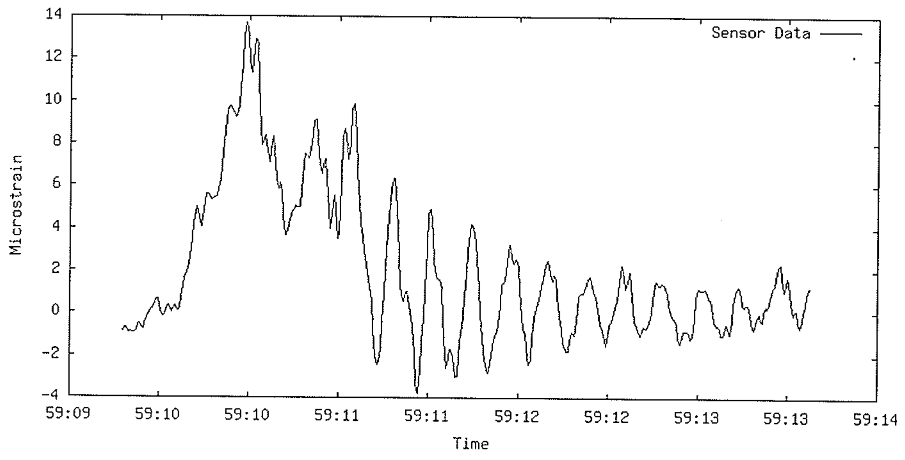

Figure 4.2: Example event in "Normal Truck" class

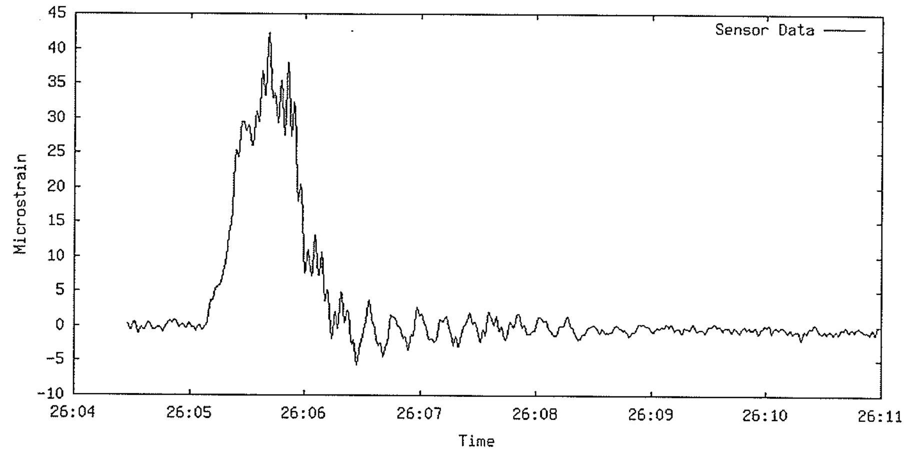

Figure 4.3: Another example event in "Normal Truck" class

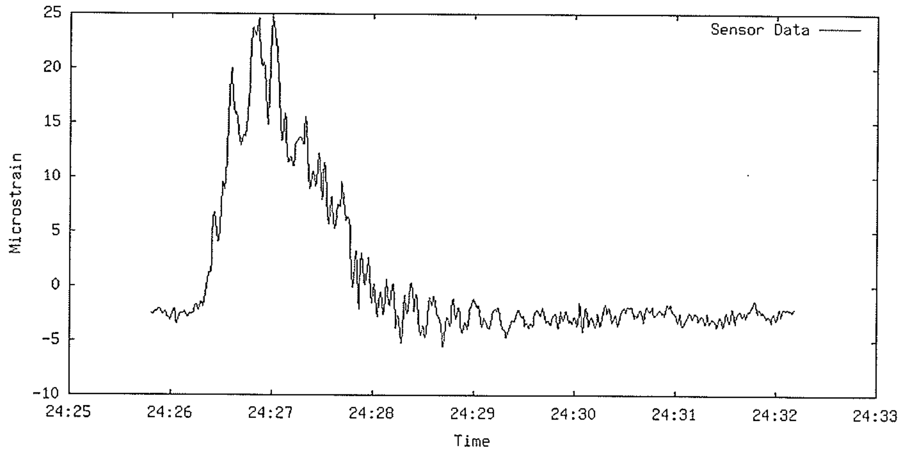

Figure 4.4: Example event in "Large Truck" class

25. The two single trailer trucks have peak microstrains of about 14 and 42. While these graphs are artificially chosen to represent the difficulty of the problem, they are not atypical. Also, while these figures might suggest that the duration of the event is a good measure of the class, similar variation can be seen in event duration as well.

While most of the features chosen demonstrated a clear difference in the mean feature value by class, the within-class variation of the feature values was much larger than the inter-class spacing of mean values. This means that there was significant overlap between the value of a particular feature in different classes, and choosing a threshold value to minimize misclassification becomes a difficult problem.

Part of the problem might be that there is so much natural variation in the actual vehicles being observed. The class of "Other Vehicles" represents both passenger trucks towing empty animal trailers and fully loaded cement trucks. It also represents busses, which have a similar length to semi-trailer trucks. The weight of an empty semi-trailer truck and a fully loaded truck are likely extremely different. In addition, various suspension types and trailer types influence the application of the load over time.

It is therefore easy to imagine how distinguishing an empty dual-trailer flatbed truck and a partially-loaded cement truck might be a difficult task. While it may be fairly easy to define thresholds to classify the "typical" example of each class, the training data is extremely limited in covering all possible combinations of vehicle type, trailer type, suspension type, vehicle speed, and vehicle loading.

Despite these challenges, automatic classification was still attempted. However, it is important to note that even an ideal classification algorithm would be inherently limited by the limitations of the data presented to it.

4.2 Background

Neural networks are a method of generalizing a function from a limited set of points. They are loosely based on the structure of biological neurons in the brain. In its most general form, such a network consists of a number of artificial neurons, and a number of adaptable weighted connections between these neurons.

Two types of neural networks are used in this thesis, a multi-layer perceptron network and a radial basis function network.

4.2.1 Multi-Layer Perceptrons

The multi-layer perceptron is a network composed of two or more layers of weights. The first layer of weights connect the inputs to a layer of hidden neurons. These neurons are referred to as hidden because they exist in between the inputs and the outputs. The hidden units are connected by a second layer of weights to either more layers of hidden neurons or the output neurons. These output neurons are connected directly to each output. While there are many possible ways to arrange these connections, the most commonly used is to simply allow a fully connected network. For example, in a fully connected two-layer network, every input is connected to every hidden unit, and every hidden unit is connected to every output unit. This is the case used in this thesis.

For a multi-layer perceptron, the neurons are very simple. The connections between the hidden units and the inputs simply compute a weighted sum of the inputs. A non-linear activation function is then applied to this sum. Examples of possible activation functions include the Heaviside function, the *tanh* function, and the logistic sigmoid. The activation function allows the multi-layer perceptron to model general non-linear transformations. In this thesis, the logistic sigmoid is used as the transfer function due to it's computational simplicity. It has the following form:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{4.1}$$

In a two-layer perceptron, the weights between the hidden units and the output units compute another weighted sum, this time for each of the hidden units. While another activation function can be applied here if desired, this thesis simply uses a linear activation function.

The network is trained by applying a set of inputs, and calculating an error function of each output value from a target value. The derivative of this error function is then calculated for each of the output neurons. This error is then transferred back to each hidden neuron of the network according to how much influence it had on the result. The weights are then adjusted in the direction that will reduce the error, by some amount that is equal to the magnitude of the error times some learning rate. The learning rate parameter controls how quickly the network adapts to an input.

The method of error back-propagation is an efficient way of making the network gradually learn to provide a certain output when a certain input is applied. Since the form of the network is extremely flexible, it is possible to learn and generalize many such mappings at once.

4.2.2 Radial Basis Functions

Radial basis functions are similar in many ways to multi-layer perceptrons. The major difference is that the first layer of weights and the hidden neurons are replaced by a set of vectors, called basis functions.

In a radial basis function network, the outputs of the first layer are based solely on some function of the distance between a basis vector and an input vector in the input space. In this thesis, the gaussian function is used:

$$f(x) = e^{\frac{x^2}{2\sigma^2}}$$

In this formula, x represents the distance between the input vector and the basis vector. The value of σ is a parameter of the system.

The second layer of the radial basis function network used in this thesis is simply a linear network of weights, exactly like that used to connect the hidden units to the output units in the previous section.

In a radial basis function network, the first and second layers are trained differently. The second layer is trained using back-propagation as described before. However, the first layer is trained using the inputs only, with no knowledge of the target vector. Basically, based on a particular input vector, the basis vectors are moved slightly in order that one or more of them might better represent the input vector being applied. There are a number of ways to do this, including frequency sensitive competitive learning and soft competitive learning.

The RBF network in this thesis uses soft competitive learning. In this method, all of the basis vectors are moved proportionally to their activation in the direction of the input vector. In this way, the basis vectors that are close to the input vector will move the most, since they will have the greatest activation, while vectors which are more distant will be largely unaffected. The purpose of this training is to cause the basis vectors to move to cover the areas of the input space where the inputs are at the greatest concentration. The second layer then learns a linear function to map basis function activations to output vectors. Training on both layers involves an adjustable learning rate parameter which affects the degree to which the weights and basis vectors change.

4.3 Neural Network Implementation Details

For this thesis, both a multi-layer perceptron and a radial basis function network are used to learn a mapping from the feature values to the vehicle classes. These networks function as described in sections 4.2.1 and 4.2.2 respectively. However, a few implementation details should be discussed.

First of all, the values of the adaptive weights in both networks are initially assigned a random value between -0.5 and 0.5. The basis vectors of the RBF are initially set to be equal to a randomly chosen input vector.

Secondly, the training inputs are analyzed for their mean and standard deviation. Each value applied to the network then has the training mean subtracted and is divided by training standard deviation. This is done to normalize the inputs. It is possible, for example, that one raw input has a large variance while another has a tiny variance. This would cause the neural networks to put less weight on the contributions of the second input. In particular, RBF networks require normalized inputs, since they use only the distance in input space as their first layer mapping.

Next, the training inputs are applied one at a time to the network. Once all of the inputs have been applied, they are randomly shuffled and then applied again. This process repeats until a predetermined number of training cycles have been performed. The shuffling is important because with large learning rates, the last few inputs applied at any given cycle would have a greater effect on the result. Shuffling means that while a particular input might have more influence at the end of one cycle, it will be in a random location in the following cycle.

In addition, the learning rate parameter is gradually reduced during the training process. This reduction is linear, and it is set so that the first cycle is performed with the initial learning rate, and the cycle after the final cycle would be done with a learning rate of zero. For example, with 10 cycles and an initial learning rate of 0.1, the first cycle would have a learning rate of 0.1, and the final cycle would have a learning rate of 0.01, with reductions of 0.01 after each cycle. This is done so that the system can adapt quickly at first, and when it has settled into the general area of a good mapping, it can refine its output with smaller and smaller steps until training is completed.

In both neural networks, each vehicle class is represented by a single output unit. The target vector is set to one if the output represents the correct class for the inputs, and zero if it does not. The final classification decision is made by choosing the output unit with the highest activation.

There are a number of adjustable parameters which must be selected for each network. The selection of these values is the topic of the following section.

4.4 Neural Network Optimization

The optimization of the neural networks were done in two steps. First of all, the adjustable parameters of the networks are varied with a fixed set of input features. Then, the input features are varied with the optimal parameters identified during the first search. Separate optimizations were done for the perceptron network and the RBF network.

All optimization was done on the problem of classifying the entire February data set using the entire August data set as training. The full data sets were used in order to provide the best resolution of results for comparison, and cross-seasonal testing was done to ensure that the results were not specific to a particular season. Please note that this process will create a bias in the results. Since the parameters and features were chosen to optimize this particular problem, it is likely that when the networks are tested later that networks trained with August data and tested with February data will have an advantage over other networks.

Two sub-problems are investigated in the feature search: a classification into 8 classes, and a classification into 4 classes. In all following discussions, the 8 class problem includes the following classes (shorthand names in brackets):

1. Unobserved events (XC)
2. "Other vehicle" events, left lane (SL)
3. "Other vehicle" events, right lane (SR)
4. Single trailer trucks, left lane (NL)
5. Single trailer trucks, right lane (NR)
6. Multi-trailer trucks, left lane (LL)
7. Multi-trailer trucks, right lane (LR)
8. Multiple vehicle events (MP)

The 4 class problem includes only the following classes:

1. All other events (XC)
2. Single and multi-trailer trucks, left lane (NL)
3. Single and multi-trailer trucks, right lane (NR)
4. Multiple vehicle events (MP)

Additionally, in the following sections all trials were repeated 10 times, and the average hit rate is presented. The hit rate refers to the percentage of events which were properly classified by the system. Repeated trials were performed in order to minimize variation due to initial conditions.

When features are referred to, they are specified by integer ID numbers for the sake of formatting. Please refer to Table 2.1 for a description of each feature number.

4.4.1 Parameter Search

The adjustable parameters of the network include the number of training cycles, the number of hidden nodes or basis functions, and the initial learning rate. In addition, the RBF networks also have the gaussian width parameter, σ. Three values were selected for each of these parameters, and an exhaustive search of all combinations was performed.

The number of training cycles influences how rigidly the system conforms to the input. Networks can be overtrained in the sense that they match the training data very well but do not generalize well to other sets of data. The number of hidden nodes or basis functions influences the flexibility of the system. Too many nodes provides a system that is too flexible and will conform to the training data, but will not generalize well to other data. Too little flexibility hinders the ability of the system to match complex mappings.

The initial learning rate works with the number of training cycles to adjust how quickly the system learns. The gaussian width controls the size of the basis functions, and should be adjusted along with the number of basis functions to allow the system to cover all the data in the input space well.

For the following tests, 17 input features were used. The IDs of these features are 4, 7, 8, 10, 16, 19, 25, 27, 29, 30, 31, 34, 35, 36, 37, 38, and 39. These include a variety of all of the feature types discussed in section 2.5.

The following values were tested:

- Training cycles: 1000, 2000, and 4000 cycles
- Hidden nodes/Basis Functions: 20, 40, and 80 nodes/vectors
- Initial learning rates: 0.01, 0.02, and 0.04
- Gaussian Width: 0.05, 0.1, and 0.2 standard deviations

Testing all of the combinations of parameters resulted in 27 trials for the perceptron network, and 81 trials for the RBF network. The full results are not shown here, but variation across the trials was very small.

For the perceptron, the best results were found with 1000 training cycles, 80 hidden nodes, and an initial learning rate of 0.01. Since all of these values are the maximum and minimum tested, smaller variations were tested as well from this ideal. The tested variations were plus and minus 100 training cycles, plus and minus 10 hidden nodes, and plus and minus 0.002 learning rate. No significant improvement was found, so this was chosen as the ideal network parameters for the perceptron network.

The best results for the RBF network were found with 2000 training cycles, 80 basis functions, an initial learning rate of 0.01, and a gaussian width of 0.2. Once again, the same smaller variations around the extreme values, as well as plus and minus 0.01 gaussian width, were tested with no significant improvement. However, the hit rate for a network with 2000 training cycles was very close (less than 0.01% difference) to the network with 4000 training cycles, so a compromise of 3000 training cycles was tested with the expectation that the peak lies somewhere between the two values, and was found to be better. The ideal parameters for the RBF network were therefore chosen to be 3000 training cycles, 80 basis functions, 0.01 initial learning rate, and 0.2 standard deviations for the gaussian width.

4.4.2 Feature Search

The search of the possible feature space uses an algorithm described in [6] called "forward selection search procedure". It is a method used in regression analysis to select a set of variables to include in a model. Basically, it starts with a model of only one variable, and calculates the fitness of the model with each possible variable. The best variable is chosen, and then it tests two variable models with the chosen variable and every other variable. If one or more of these variables improve the model, then the best variable is added and the algorithm continues. The process stops when no variable can be found to add to the model that improves it.

For this thesis, the variables to be chosen are the features from section 2.5. The fitness criteria will be the average hit rate across ten trials, as described in the previous sections. Tables 4.1 and 4.2 show the feature search results for the perceptron network and the RBF network, respectively. These values are optimized to solve the problem of classifying the data into 8 categories. Please use Table 2.1 for a discussion of the meaning of these feature IDs.

Please note that the algorithm does not add features indefinitely, since there comes a point where the penalties of a new feature exceed the benefits. In particular, each new feature adds to the dimensionality of the input space. With a fixed amount of training data, this means that the input space is covered more sparsely by the training data points, which means that the resulting mappings will be less clearly defined.

Some interesting information can be found by analyzing these results. For instance, when adding a feature causes the incremental improvement of another feature to decrease, it means that the two features contain redundant information, and so when one of them is added, the usefulness of the other one is decreased. On the other hand, when adding one feature causes the incremental improvement of another feature to increase, that implies that there is some information contained in the combination of the two features that isn't apparent when the features are examined alone. Examples of this phenomenon can be seen in feature 14 after feature 39 is added to the perceptron network, and in feature 24 after feature 11 is added to the RBF network.

The results of the algorithm for the 8 class problem are as follows:

- Perceptron network: Features 4, 10, 12, 14, and 39
- RBF network: Features 4, 8, 11, and 24

Additionally, the same test was applied to the simpler problem of classifying vehicles into only 4 classes. The results of these tests can be seen in Tables 4.3, 4.4, and 4.5. The ideal features for these problems were found to be:

- Perceptron network: Features 4, 7, 13, 14, 15, 23, 24, 32, and 34
- RBF network: Features 4, 8, and 34

Table 4.1: Feature search results for perceptron with 8 classes

Input	Base Sensors					
	None	4	4,12	4,12,10	4,12,10,39	4,12,10,39,14
3	21.72%	-15.29%	-27.82%	-35.69%	-34.27%	-42.12%
4	**41.13%**	N/A	N/A	N/A	N/A	N/A
5	21.34%	0.20%	-6.04%	-4.59%	-5.14%	-5.64%
6	21.95%	-3.81%	-7.93%	-6.30%	-5.05%	-5.06%
7	33.02%	4.45%	-6.83%	-7.38%	-6.51%	-6.16%
8	34.94%	5.90%	-3.66%	-5.66%	-4.13%	-4.27%
9	28.63%	8.66%	4.70%	-0.61%	-0.11%	-0.79%
10	25.46%	6.86%	**5.24%**	N/A	N/A	N/A
11	28.48%	6.30%	-2.52%	-2.10%	-1.65%	-3.32%
12	31.74%	**11.72%**	N/A	N/A	N/A	N/A
13	28.92%	6.55%	-0.35%	-0.44%	1.33%	-0.79%
14	29.79%	4.39%	1.40%	-0.37%	**1.77%**	N/A
15	29.42%	7.09%	4.09%	-1.48%	-0.41%	-1.39%
16	25.35%	6.34%	3.99%	0.06%	0.99%	-0.05%
17	26.46%	-7.87%	-2.12%	-2.85%	-1.97%	-0.96%
18	23.32%	1.92%	-0.09%	0.53%	0.64%	-1.13%
19	26.23%	0.44%	1.83%	0.37%	0.84%	-1.49%
20	26.34%	5.50%	-2.88%	-0.93%	-0.76%	-2.04%
21	23.32%	-0.02%	-0.18%	0.26%	0.52%	-1.30%
22	23.32%	1.01%	-0.15%	0.35%	0.55%	-1.45%
23	23.32%	0.29%	-0.08%	0.27%	0.56%	-1.65%
24	26.68%	-3.05%	2.39%	-0.29%	0.14%	-1.80%
25	23.32%	1.48%	-0.11%	0.23%	0.70%	-1.37%
26	23.32%	1.02%	0.84%	0.05%	0.40%	-0.76%
27	31.14%	11.59%	-0.64%	0.20%	0.76%	-0.98%
28	23.32%	1.39%	-0.32%	0.27%	0.24%	-1.59%
29	24.39%	-3.67%	-5.73%	-7.62%	-5.35%	-5.35%
30	24.12%	-0.34%	-0.81%	0.02%	0.15%	-1.49%
31	24.05%	-2.82%	-8.26%	-7.26%	-6.49%	-6.30%
32	30.53%	-3.37%	-6.77%	-9.74%	-7.87%	-8.54%
33	28.43%	-3.25%	-6.77%	-6.46%	-4.70%	-5.93%
34	35.59%	3.52%	-0.08%	-2.41%	-4.76%	-4.04%
35	39.39%	3.87%	1.46%	-2.96%	-3.54%	-2.20%
36	26.68%	1.94%	-2.01%	-0.90%	-1.42%	-2.15%
37	37.73%	0.27%	0.08%	0.30%	-0.87%	-1.91%
38	22.04%	-4.10%	-6.54%	-1.31%	-3.60%	-1.37%
39	22.41%	2.07%	1.68%	**0.90%**	N/A	N/A

Table 4.2: Feature search results for RBF network with 8 classes

Input	Base Sensors				
	None	8	8,4	8,4,11	8,4,11,24
3	26.83%	-7.09%	-19.01%	-21.81%	-24.65%
4	24.24%	**11.92%**	N/A	N/A	N/A
5	17.53%	-2.12%	-3.75%	-2.26%	-6.11%
6	20.12%	-1.13%	-5.00%	-3.99%	-9.21%
7	33.48%	1.16%	0.70%	1.43%	-2.12%
8	**33.92%**	N/A	N/A	N/A	N/A
9	23.32%	-0.63%	-1.48%	1.14%	-1.51%
10	23.32%	-0.53%	-6.68%	-0.96%	-3.48%
11	29.45%	10.00%	**2.80%**	N/A	N/A
12	23.32%	-1.01%	-3.03%	-0.56%	-3.96%
13	23.32%	-0.23%	-3.19%	-1.34%	-4.01%
14	23.32%	-0.72%	-2.70%	-0.88%	-1.19%
15	25.76%	0.29%	-3.58%	1.62%	-0.50%
16	23.32%	-0.72%	-3.23%	1.01%	-2.65%
17	23.34%	-0.26%	-2.73%	-2.97%	-5.00%
18	23.32%	-0.95%	-2.68%	0.76%	-0.63%
19	23.32%	-1.01%	-2.44%	-0.20%	-1.05%
20	22.82%	-0.08%	-3.05%	-0.49%	-0.85%
21	23.32%	-1.20%	-3.06%	1.31%	-1.34%
22	23.32%	-0.93%	-2.65%	-0.03%	-1.55%
23	23.32%	-0.78%	-2.97%	-0.73%	-2.36%
24	23.32%	-0.37%	-2.27%	**2.84%**	N/A
25	23.32%	-1.04%	-4.01%	-1.66%	-1.20%
26	23.32%	-0.40%	-4.60%	0.02%	-3.64%
27	23.32%	-0.85%	-2.77%	0.43%	-3.57%
28	23.32%	-1.04%	-2.21%	0.79%	-3.52%
29	22.39%	1.54%	-1.89%	-1.19%	-4.79%
30	24.09%	0.20%	-3.37%	2.01%	-2.04%
31	24.44%	0.52%	-2.67%	-1.71%	-4.77%
32	25.24%	2.30%	0.88%	0.35%	-5.09%
33	21.28%	0.37%	-1.11%	-1.43%	-5.49%
34	15.35%	0.40%	1.66%	1.51%	-2.16%
35	28.55%	7.93%	-0.11%	-3.20%	-4.66%
36	25.00%	1.42%	-2.76%	-1.19%	-3.05%
37	28.20%	4.48%	-0.87%	-3.22%	-4.50%
38	29.13%	1.30%	-4.38%	-3.81%	-6.63%
39	33.90%	6.19%	-0.11%	-0.85%	-1.78%

Table 4.3: Feature search results for perceptron with 4 classes - Part 1

Input	Base Sensors					
	None	34	34,13	34,13,15	34,13,15,24	34,13,15,24,32
3	27.59%	-38.90%	-41.39%	-49.13%	-54.16%	-54.85%
4	63.69%	3.69%	2.50%	0.21%	-0.72%	-0.05%
5	39.27%	-1.66%	-5.61%	-0.26%	-1.92%	-2.04%
6	36.81%	-1.49%	-1.71%	-0.87%	-0.76%	-0.40%
7	44.57%	3.99%	3.11%	0.08%	-0.58%	**0.63%**
8	43.67%	4.33%	3.66%	-0.30%	-1.07%	-1.37%
9	36.28%	3.64%	4.16%	0.09%	-0.44%	0.27%
10	56.63%	0.72%	3.89%	0.27%	-0.64%	-0.44%
11	58.77%	-1.55%	-4.97%	-1.94%	-2.65%	-2.45%
12	61.19%	2.58%	-2.61%	0.26%	-0.44%	-0.24%
13	58.25%	**4.95%**	N/A	N/A	N/A	N/A
14	51.68%	-0.50%	-0.91%	-0.05%	-0.06%	0.46%
15	57.85%	4.86%	**5.67%**	N/A	N/A	N/A
16	57.00%	0.17%	3.54%	-0.27%	-0.52%	-0.21%
17	53.89%	-0.85%	1.49%	-0.03%	-0.99%	-0.50%
18	27.59%	-0.18%	-0.46%	0.20%	-0.26%	-0.06%
19	59.28%	2.15%	2.77%	0.43%	0.00%	0.08%
20	57.44%	-3.70%	-4.73%	-3.40%	-3.87%	-3.45%
21	27.59%	-0.11%	-1.33%	-0.11%	-0.23%	-0.24%
22	27.59%	-0.11%	-1.75%	-0.03%	-0.27%	0.20%
23	27.59%	-0.14%	-1.20%	0.27%	-0.20%	-0.14%
24	31.54%	1.92%	2.26%	**0.91%**	N/A	N/A
25	27.59%	-0.18%	-1.37%	-0.12%	-0.29%	0.18%
26	53.69%	-0.47%	0.35%	-0.18%	-0.14%	-0.17%
27	60.46%	2.13%	-3.08%	0.14%	-0.56%	-0.24%
28	27.59%	-0.18%	-2.09%	-0.02%	-0.24%	-0.02%
29	39.68%	-0.55%	-2.03%	0.75%	0.21%	0.27%
30	28.43%	-0.02%	-1.40%	-0.20%	-0.64%	-0.43%
31	28.31%	-0.72%	-3.29%	0.00%	-0.40%	-2.03%
32	49.77%	-0.56%	0.32%	0.35%	**0.23%**	N/A
33	41.60%	0.26%	0.78%	-0.23%	-0.14%	-0.12%
34	**64.02%**	N/A	N/A	N/A	N/A	N/A
35	62.65%	-1.25%	-2.79%	0.38%	-0.76%	-1.43%
36	45.35%	-1.17%	-3.60%	-0.53%	-1.27%	-1.28%
37	55.02%	-1.28%	-1.11%	0.75%	-0.85%	-0.18%
38	44.82%	-0.88%	-0.41%	-0.56%	-1.74%	-1.05%
39	58.84%	-0.75%	-0.20%	0.34%	-0.78%	-0.34%

Table 4.4: Feature search results for perceptron with 4 classes - Part 2

Input	Base Sensors			
	34,13,15,24,32,7	Adding 4	Adding 14	Adding 23
3	-65.15%	-48.28%	-53.83%	-50.76%
4	**0.50%**	N/A	N/A	N/A
5	-1.27%	0.30%	0.18%	-0.21%
6	-0.53%	-0.08%	-0.02%	-0.93%
7	N/A	N/A	N/A	N/A
8	-2.30%	-1.68%	-2.04%	-2.59%
9	-0.30%	0.55%	0.05%	-0.79%
10	0.09%	0.73%	-0.06%	-0.30%
11	-0.63%	-1.19%	-2.09%	-2.03%
12	-0.06%	0.29%	-0.27%	-0.88%
13	N/A	N/A	N/A	N/A
14	-0.49%	**1.04%**	N/A	N/A
15	N/A	N/A	N/A	N/A
16	-0.15%	0.63%	0.21%	-0.21%
17	-0.52%	0.47%	-0.90%	-0.58%
18	-0.18%	0.18%	0.21%	-0.61%
19	-0.15%	0.21%	-0.08%	-0.88%
20	-3.20%	-1.94%	-2.68%	-3.48%
21	-0.17%	0.05%	-0.12%	-0.59%
22	-0.32%	0.49%	-0.23%	-0.58%
23	-0.30%	0.38%	**0.24%**	N/A
24	N/A	N/A	N/A	N/A
25	-0.35%	0.18%	-0.75%	-0.72%
26	-0.18%	0.61%	-0.15%	-0.78%
27	-0.47%	0.20%	-0.37%	-0.72%
28	0.03%	0.67%	-0.30%	-0.75%
29	0.26%	0.05%	-0.75%	-1.23%
30	-0.56%	0.02%	-0.69%	-1.02%
31	-2.56%	-1.04%	-1.48%	-2.04%
32	N/A	N/A	N/A	N/A
33	-0.46%	-0.87%	-1.60%	-1.77%
34	N/A	N/A	N/A	N/A
35	-1.57%	-0.73%	-1.55%	-1.63%
36	-1.66%	-0.81%	-0.95%	-1.11%
37	-1.25%	-0.15%	-0.27%	-0.61%
38	-0.75%	-0.41%	-1.42%	-1.78%
39	-0.24%	0.18%	-0.14%	-0.27%

Table 4.5: Feature search results for RBF network with 4 classes

Input	Base Sensors			
	None	4	4,8	4,8,34
3	49.54%	-11.43%	-15.15%	-24.04%
4	**60.98%**	N/A	N/A	N/A
5	46.65%	-3.02%	-5.99%	-6.13%
6	31.86%	-9.88%	-2.38%	-3.61%
7	43.17%	-16.57%	-1.77%	-1.04%
8	41.36%	**3.72%**	N/A	N/A
9	27.59%	0.26%	-6.11%	-0.81%
10	27.74%	-5.17%	-5.15%	-1.52%
11	35.12%	-6.25%	-2.16%	-1.98%
12	27.59%	-0.15%	-5.30%	-1.16%
13	49.54%	-0.15%	-5.18%	-0.84%
14	27.59%	0.00%	-4.63%	-1.22%
15	30.49%	-33.20%	-6.84%	-1.01%
16	27.59%	-0.30%	-3.06%	-0.58%
17	53.20%	-4.53%	-3.51%	-1.14%
18	27.59%	0.00%	-4.88%	-1.19%
19	27.59%	0.00%	-4.74%	-1.04%
20	54.36%	-1.36%	-4.09%	-0.24%
21	27.59%	0.00%	-5.06%	-1.17%
22	27.59%	0.00%	-5.26%	-1.07%
23	27.59%	0.00%	-2.45%	-1.59%
24	27.59%	-0.11%	-3.84%	-1.20%
25	27.59%	0.00%	-5.03%	-1.25%
26	27.59%	-0.24%	-3.45%	-1.63%
27	27.65%	-0.24%	-6.28%	-1.13%
28	27.59%	0.00%	-6.05%	-0.34%
29	37.94%	-12.87%	-2.55%	-3.63%
30	28.35%	0.15%	-3.90%	-1.16%
31	29.28%	0.88%	-1.86%	-2.97%
32	30.98%	-5.88%	2.97%	-2.65%
33	25.41%	-4.62%	-5.46%	-1.49%
34	26.57%	-0.95%	**8.89%**	N/A
35	35.56%	-2.39%	6.39%	-3.48%
36	45.73%	-12.94%	1.17%	-4.21%
37	53.96%	-0.06%	5.21%	-2.91%
38	44.79%	-0.59%	4.28%	-1.54%
39	53.51%	0.58%	5.90%	-2.90%

Table 4.6: Full results from all neural network tests

		Aug Training		Feb Training	
Problem	Network	Aug Test	Feb Test	Aug Test	Feb Test
8 classes	Perceptron	59.84%	52.21%	63.93%	63.24%
8 classes	RBF	64.75%	44.85%	45.08%	46.32%
4 classes	Perceptron	86.01%	80.15%	77.87%	81.62%
4 classes	RBF	82.79%	67.65%	63.11%	68.38%

It is interesting to note that the RBF networks seem to require fewer features. This is probably due to the fact that RBF networks pay a higher penalty for a sparsely covered input space, since they need to have basis functions covering all clusters of data points.

4.5 Neural Network Results

In order to properly evaluate the accuracy of the neural networks in various situations, the August and February data sets were split into training and test data sets. In the August data, 122 events were chosen at random for the test set, leaving 459 events in the training set. In the February data, 136 events were chosen for the test set and 520 were left in the training set.

The four neural networks identified in the previous section were tested on four combinations of training and test data. The results are presented in Table 4.6.

Some interesting results are apparent here. First of all, the bias discussed in section 4.4 is readily apparent in the RBF networks, but not in the perceptron networks. For both of the RBF trials, the August training data with the August test data had a significant lead over the other data set combinations. This is probably because the network parameters and features were chosen to represent the August data. Therefore, the basis functions could easily cover the August data, and could classify this data well. However, when other data sets were used for either training or testing, the network did not perform as well.

The data from the perceptron network is also interesting. With the exception of the 8 class February training, August testing trial, there is a sig-

Table 4.7: Loss matrix: 8 classes with perceptron network

	XC	SL	SR	NL	NR	LL	LR	MP
XC	13	0	21	4	5	0	0	0
SL	0	0	5	1	0	0	0	0
SR	1	0	12	1	4	0	0	0
NL	0	0	1	16	1	0	0	1
NR	0	0	5	1	28	0	0	0
LL	0	0	0	0	0	0	0	1
LR	0	0	0	0	4	0	0	0
MP	0	0	2	2	5	0	0	2

nificant drop in the hit rate when using a set of test data from the opposite season as the training data. This, along with the problems with the RBF networks, seems to suggest that there are significant changes in the feature space between seasons.

In conclusion, the perceptron network seems to be more resistant to seasonal changes, although it could also benefit from periodic recalibration in order to reduce errors.

4.5.1 Loss Matrices

The loss matrix is a common tool for analyzing the results of a classification algorithm. Basically, it is a table which shows the actual class of an event versus the class chosen by the classifier. The actual classes are listed along the side, while the chosen classes are listed along the top. Each column represents a chosen class, while each row represents an actual class. Events in the main diagonal of this matrix are properly classified, while events off of the main diagonal are not. Analysis of the confusion matrix can show where misclassifications are occurring.

For the sake of formatting, the shorthand described in section 4.4 is used to label the classes. Loss matrices are shown for the August training data and the February test data, for all four types of network (Tables 4.7, 4.8, 4.9, and 4.10).

In the 8 class problem, it seems that the system resists classifying any events as multi-trailer trucks or "other vehicles" in the left lane. This is

Table 4.8: Loss matrix: 8 classes with RBF network

	XC	SL	SR	NL	NR	LL	LR	MP
XC	11	0	15	15	2	0	0	0
SL	0	0	1	5	0	0	0	0
SR	2	0	11	1	4	0	0	0
NL	0	0	0	16	2	0	0	1
NR	7	0	2	2	22	0	0	1
LL	0	0	0	0	0	0	0	1
LR	0	0	0	0	4	0	0	0
MP	1	0	0	5	4	0	0	1

Table 4.9: Loss matrix: 4 classes with perceptron network

	XC	NL	NR	MP
XC	56	7	4	0
NL	0	16	2	2
NR	3	1	34	0
MP	1	4	3	3

Table 4.10: Loss matrix: 4 classes with RBF network

	XC	NL	NR	MP
XC	49	9	9	0
NL	1	15	2	2
NR	8	2	26	2
MP	4	3	2	2

probably due to the extremely low frequency of these events in the data. Even when the input features are in the perfect space for one of these classes, it is still more likely that the event is in a more common class, such as unobserved events or single-trailer trucks. Unfortunately, there appears to be no easy solution to this problem with the features and data that are available.

The results from the 4 class loss matrices are much better. With the reduction in classes, all of the possible classes have plenty of training examples.

4.6 KDE Weight Analysis using Neural Network Classifications

In section 3.4 an attempt was made to estimate a vehicle's weight by looking at the kernel density estimate of event intensity. The event intensity was defined as the average of the peak microstrain measurement across all 9 girder ESG sensors. It may be possible to refine this calculation by using the 4 class perceptron classifier that was just tested. This classifier was found to have a valid classification rate of over 80% over the four classes.

Such a network could be used to split semi-trailer trucks out from the other vehicle and multiple vehicle classes. In addition, it will differentiate between trucks in the left and right lanes. Since it seems probable that the measured event intensity will differ with the lane of the vehicle due to proximity to the various sensors, this should improve the accuracy of the KDE.

The four improved KDEs are shown in Figures 4.5, 4.6, 4.7, and 4.8. The results for multiple-vehicle events and non-truck events are interesting, particularly in showing the huge variety of event intensities being classified in each of these categories. This indicates that the neural network is using more complex parameters than simply the size of the event in order to determine the class label.

The most useful graphs, however, are Figures 4.6 and 4.7. With these, the magnitude of a fully loaded vehicle event can be estimated. It seems reasonable to assume that most trucks will be below the maximum allowed gross vehicle weight of 62500 kg. By looking at the graph, a value can be visually chosen for each lane that includes the majority of the trucks.

If the assumption is made that a small portion of the vehicles identified are actually overweight, a safe choice for the fully loaded vehicle intensity seems

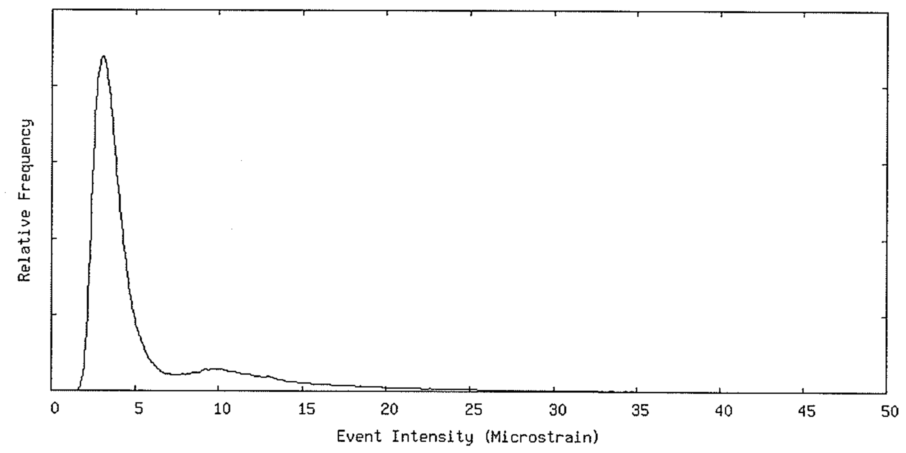

Figure 4.5: KDE of non-truck events

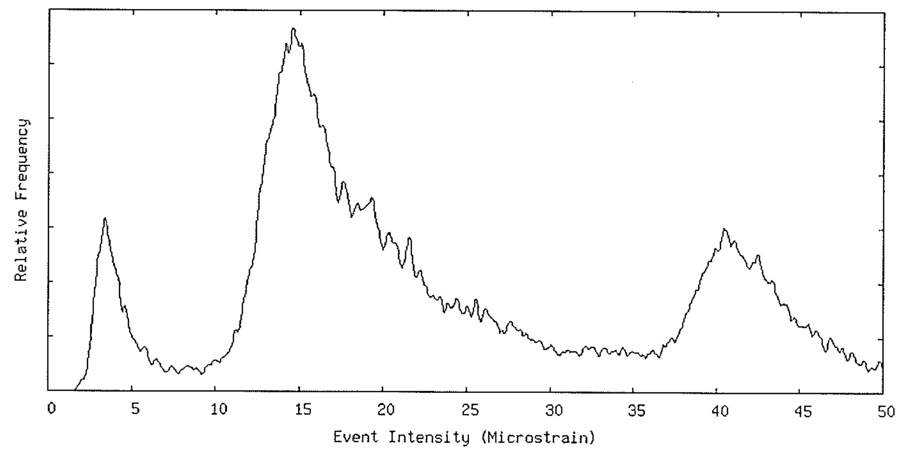

Figure 4.6: KDE of trucks in left lane

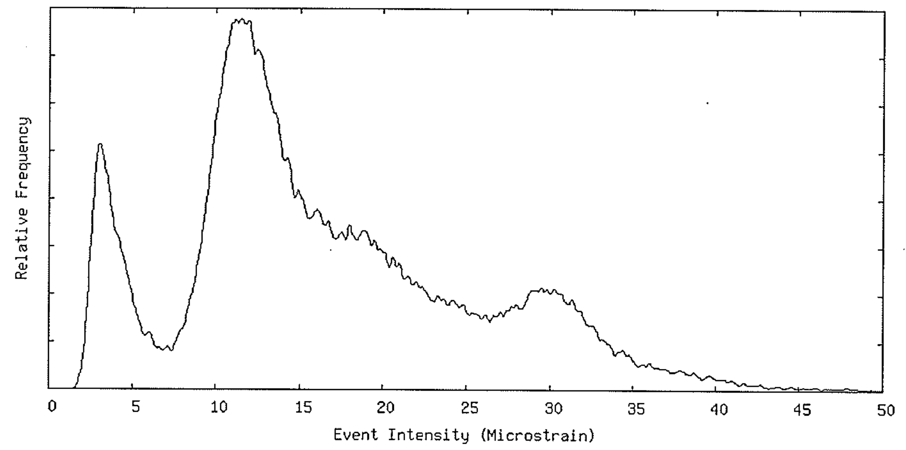

Figure 4.7: KDE of trucks in right lane

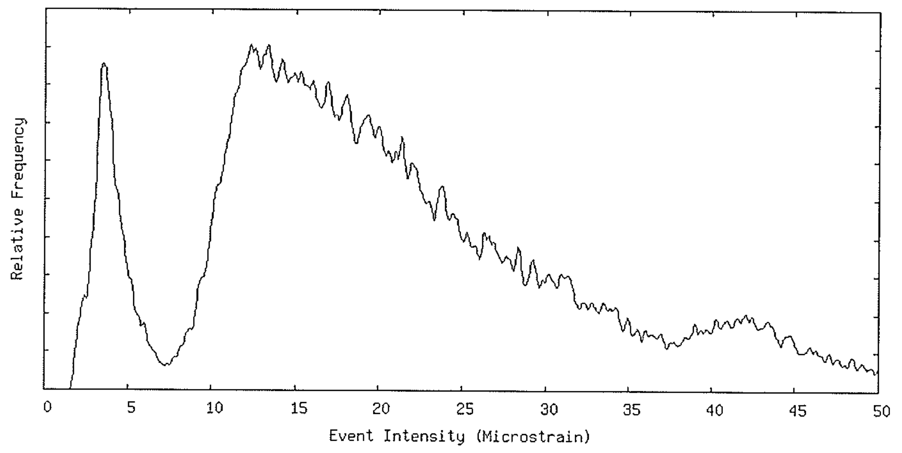

Figure 4.8: KDE of multiple vehicle events

Table 4.11: Event intensity to microstrain conversions

	KDE Est.	Small Truck	Large Truck
Left Lane (kg/microstrain)	1302	903	1159
Right Lane (kg/microstrain)	1563	1201	1730

to be about 48 microstrain in the left lane and 40 microstrain in the right lane. Using the maximum vehicle weight of 62500, this gives a conversion of 1302 kg/microstrain for the left lane and 1563 kg/microstrain for the right lane, if linear conversion is used.

Referring to the weighed vehicles from the July data in Table 3.1, it is possible to calculate an average event intensity of 33.587 microstrain for the smaller truck in the right lane, versus 44.668 microstrain in the left lane. Using its weight of 40340 kg, conversions of 1201 kg/microstrain in the right lane and 903 kg/microstrain in the left lane are calculated. Similarly, the large truck has average intensities of 36.863 and 55.020 for the right and left lanes, respectively. This translates to conversions of 1730 kg/microstrain and 1159 kg/microstrain for the right and left lanes. These results are summarized in Table 4.11.

While these results are not perfect, they are a significant improvement over the results in section 3.4. It appears that either significant variation in the event intensity with the July data or non-linearities in the conversion prevent further refinement.

Chapter 5

Conclusions

This thesis has shown a number of useful results. A novelty detection algorithm has been created and has been shown to consistently identify vehicle events over a long period of time. This novelty detection algorithm even appears to be sensitive enough to identify passenger vehicles under some conditions. Interesting patterns in the number and intensity of these events over time have been found and discussed.

Further, a number of features of an event have been developed and analyzed for the classification of these vehicle events. The use of these features in order to train neural networks for classification has been investigated. The optimal network parameters and features were found for a variety of situations.

When classifying events from a different season than the training data, perceptron networks appear to work better than RBF networks. However, even perceptron networks experience a decrease in accuracy in a different season. This result implies that if such a system were to be used in practice, it would need to be recalibrated periodically to compensate for these changes.

While the networks had significant problems classifying vehicles into 8 classes, a broader classification of 4 distinct classes did show favorable results. Using this network, a KDE estimation of vehicle weights from the event intensity could be significantly improved.

5.1 Future Work

There are many directions of future work suggested by this thesis. First of all, the methods used in this thesis could be applied to other structures to see if they generalize well to different bridge models. The data gathered in this thesis was from a single structure, so it is possible that other structures and sensor networks would provide different results.

The patterns identified in the event frequency and intensity could be studied over a longer period of time in order to confirm their validity. A survey could be done to see what is occurring on the bridge when the novelty detector sees an event, in order to determine if the novelty detector is actually able to detect passenger vehicles.

Further, the classification results found here could be tested with other features in order to see if the classification rate could be improved. Also, if more data from trucks of known weight could be obtained, it would be very interesting to see if a neural network could be trained to accurately determine vehicle weights. With even a few more example vehicles of known weights, a study could be done to see if there really is a non-linear relationship between vehicle weights and peak strain measurements. Testing the KDE weight estimations obtained here against many test vehicles would improve the confidence of those results.

Finally, it would be interesting to compare the WIM results obtained here with estimates obtained using the model of the bridge. An investigation of the possibility of obtaining the weight using these parameters and a model to predict the value of these parameters under different loads could be promising. Also, further investigation into the effects of temperature and humidity on the response of the bridge could be helpful.

Bibliography

[1] United states patent 4560016: Method and apparatus for measuring the weight of a vehicle while the vehicle is in motion, 1985.

[2] United states patent 5111897: Bridge weigh-in-motion system, 1992.

[3] Christopher M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.

[4] Loren Card. Unsupervised neural computation for event identification in structural health monitoring systems. Master's thesis, University of Manioba, 2004.

[5] Thomas D. Gillespie. *Fundamentals of Vehicle Dynamics.* Society of Automotive Engineers, Inc., 1992.

[6] Kutner, Nachtsheim, Neter, and Li. *Applied Linear Statistical Models, 5th Edition.* The McGraw-Hill Companies, Inc, 2005.

[7] S.K. Leming and H.L. Stalford. Bridge weigh-in-motion system development using superposition of dynamic truck/static bridge interaction. In *Proceedings of the 2003 American Control Conference.*

[8] A. H. Memon and A. A. Mufti. Fatigue behaviour of concrete bridge deck slabs reinforced with frp and steel straps. In *5th Structural Specialty Conference of the Canadian Society for Civil Engineering.*

[9] Fred Moses. Weigh-in-motion system using instrumented bridges. *Transportation Engineering Journal*, 1979.

[10] Government of Manitoba: Infrastructure and Transportation. *Truck Weight Limit Map and Information Guide*, June 2008.

[11] U.S. Department of Transportation: Federal Highway Administration. *Traffic Monitoring Guide*, May 2001.

[12] Udo Peil. Assessment of bridges via monitoring. *Structure and Infrastructure Engineering*, 2005.

[13] John G. Proakis and Dimitris G. Manolakis. *Digital Signal Processing: Principles, Algorithms, and Applications, Fourth Edition.* Pearson Prentice Hall, 2007.