North Carolina Agricultural and Technical State University
Aggie Digital Collections and Scholarship

Dissertations | Electronic Theses and Dissertations

2015

A Novel Approach For Identifying Cloud Clusters Developing Into Tropical Cyclones

Chaunte' W. Lacewell
North Carolina Agricultural and Technical State University

Follow this and additional works at: https://digital.library.ncat.edu/dissertations

 Part of the Atmospheric Sciences Commons, Climate Commons, and the Electrical and Computer Engineering Commons

Recommended Citation

Lacewell, Chaunte' W., "A Novel Approach For Identifying Cloud Clusters Developing Into Tropical Cyclones" (2015). *Dissertations*. 103.
https://digital.library.ncat.edu/dissertations/103

This Dissertation is brought to you for free and open access by the Electronic Theses and Dissertations at Aggie Digital Collections and Scholarship. It has been accepted for inclusion in Dissertations by an authorized administrator of Aggie Digital Collections and Scholarship. For more information, please contact iyanna@ncat.edu.

A Novel Approach for Identifying Cloud Clusters Developing into Tropical Cyclones

Chaunté W. Lacewell

North Carolina A&T State University

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Department: Electrical and Computer Engineering

Major: Electrical Engineering

Major Professor: Dr. Abdollah Homaifar

Greensboro, North Carolina

2015

The Graduate School
North Carolina Agricultural and Technical State University

This is to certify that the Doctoral Dissertation of

Chaunté W. Lacewell

has met the dissertation requirements of
North Carolina Agricultural and Technical State University

Greensboro, North Carolina
2015

Approved by:

Dr. Abdollah Homaifar
Major Professor

Dr. Yuh-Lang Lin
Committee Member

Dr. Robert Li
Committee Member

Dr. Kenneth Knapp
Committee Member

Dr. John Kelly
Department Chair
Committee Member

Dr. Sanjiv Sarin
Dean, The Graduate School

© Copyright by

Chaunté W. Lacewell

2015

Biographical Sketch

Chaunté W. Lacewell was born on October 16, 1985 in Wilmington, North Carolina. She graduated with highest honors from Emsley A. Laney High School in Wilmington, North Carolina in 2003. She received her Bachelor of Science degree in Computer Engineering in 2007 and her Master of Science degree in Electrical Engineering in 2011 both with highest honors and from North Carolina Agricultural and Technical State University (NCA&TSU). In 2011, she unexpectedly pursued a Doctorate of Philosophy degree in Electrical Engineering from the same university under the supervision of Dr. Abdollah Homaifar.

Her achievements during her doctorate degree at NCA&TSU include interning with Intel Corporation in 2012 and 2013 where she received the Division Recognition Award, and accepting a full-time position at Intel Corporation. Additional achievements include publication of research papers, being a fellow of the National Science Foundation's Expedition in Computing, and being a recipient of the Wadaran L. Kennedy 4.0 Scholar Award, the University Doctoral Assistantship where she mentored students and increased the success rate for College Algebra and Trigonometry, and the Title III PhD Fellowship.

Dedication

This dissertation is the culmination of a magnificent journey. First and foremost, I dedicate this dissertation to my Savior, the Lord Jesus Christ. Through my graduate studies, I have struggled in health and have experienced many forms of discrimination but with faith, I was able to learn from the experiences and become a stronger individual. Hence, I also want to dedicate this dissertation to individuals who are experiencing difficult obstacles and are discouraged. Remember to stay focused, believe in yourself, and keep faith because with Him all things are possible.

I will also like to dedicate this dissertation to my family. A special feeling of gratitude to my loving parents Carolyn and William Lacewell Jr., my inspiring and motivational brothers Dennis and Marcus Lacewell, and LaShonda Wells. Your words of encouragement and continuous support have encouraged me throughout this process. I will always appreciate all you have done especially at times when I doubted myself and could not see the light at the end of the tunnel.

Acknowledgements

My utmost gratitude goes to my Almighty God for His strength and this life path He has given me to follow. I could not have completed this research without His guidance and I thank Him for granting me the ability to learn, strive for success, and believe in myself.

I would like to acknowledge and thank my advisor, Dr. Abdollah Homaifar, for his continuous guidance, encouragement, and support throughout my graduate studies. You have encouraged me to do my best even after I was discriminated against and you allowed me to let my intelligence, work ethic, resilience, and determination speak for itself. You showed me that my hard work will eventually pay off as it has and for that I am forever thankful. Also, the support and guidance of Dr. Kenneth Knapp from National Oceanic and Atmospheric Administration's National Climatic Data Center, my department chairperson, Dr. John Kelly, and committee members Dr. Yuh-Lang Lin and Dr. Robert Li are appreciated.

I would also like to acknowledge and thank my colleagues in the Autonomous Control and Information Technology (ACIT) center at North Carolina Agricultural and Technical State University for providing a research environment and support which were needed for the success of this research.

Table of Contents

List of Figures

List of Tables

Abstract

Providing advance notice of rare events, such as a cloud cluster (CC) developing into a tropical cyclone (TC), is of great importance. Having advance warning of such rare events possibly can help avoid or reduce the risk of damages and allow emergency responders and the affected community enough time to respond appropriately. Considering this, forecasters need better data mining and data driven techniques to identify developing CCs. Prior studies have attempted to predict the formation of TCs using numerical weather prediction models as well as satellite and radar data. However, refined observational data and forecasting techniques are not always available or accurate in areas such as the North Atlantic Ocean where data are sparse.

Consequently, this research provides the predictive features that contribute to a CC developing into a TC using only global gridded satellite data that are readily available. This was accomplished by identifying and tracking CCs objectively where no expert knowledge is required to investigate the predictive features of developing CCs. We have applied the proposed oversampling technique named the Selective Clustering based Oversampling Technique (SCOT) to reduce the bias of the non-developing CCs when using standard classifiers. Our approach identifies twelve predictive features for developing CCs and demonstrates predictive skill for 0 - 48 hours prior to development. The results confirm that the proposed technique can satisfactorily identify developing CCs for each of the nine forecasts using standard classifiers such as Classification and Regression Trees (CART), neural networks, and support vector machines (SVM) and ten-fold cross validation. These results are based on the geometric mean values and are further verified using seven case studies such as Hurricane Katrina (2005). These results demonstrate that our proposed approach could potentially improve weather prediction and provide advance notice of a developing CC by using solely gridded satellite data.

CHAPTER 1

Introduction

A tropical cyclone (TC) is a low-pressure system with closed circulation and a warm core that originates over tropical basins (Houze, 2010). These systems obtain their energy from heat fluxes from the ocean and they contain wind speeds of at least 17 m s^{-1} (Houze, 2010; Lee, 1989; Lin, 2007). The formation of TCs over the Atlantic Ocean region is an important research topic due to the lack of scientific understanding and sparse data in the area. Currently, many forecasting models are used by the National Hurricane Center to predict TCs and to prepare official track and intensity forecasts. These models run for 6-126 hours to obtain their predictions but to accurately forecast TCs, the model outputs are adjusted to match the current time and conditions. This can cause an issue if the models cannot find an exact match. Accurate models of TC development remain elusive for numerous reasons and progress in improving these models is very slow (Hennon, Helms, Knapp, & Bowen, 2011; Shen, Tao, Lau, & Atlas, 2010). In few cases, a TC could be forecasted satisfactorily in less than 24 hours prior to its development (National Oceanic and Atmospheric Administration, 2009). Satellite data are used to initialize the models and are run to attempt to forecast the complex atmospheric processes, which lead to the development of a TC. We suggest that the satellite observations along with data driven techniques could lead to an approach to identify predictive features satisfactorily that lead to the development of a TC *at least* 24 hours prior instead of using numerical weather prediction models.

TCs in the North Atlantic Ocean impact the United States; therefore, it is a research topic which needs attention to provide imperative information to citizens of the U.S. to assist in better preparedness. The 2005 Atlantic Ocean hurricane season is a fine example of this need. This

record setting hurricane season consisted of the most named TCs in history which included Hurricane Katrina which was the costliest and most expensive natural disaster in U.S. history with approximately $108 billion in damage (Dolce, 2013; McTaggart-Cowan, Deane, Bosart, Davis, & Galarneau, 2008). The impact of this hurricane season alone displays the need for better understanding of how TCs develop from cloud clusters (CCs).

The purpose of this research is summarized by one question: What determines whether a CC will develop into a TC? Forecasters have theories to answer this question from a climatology perspective, but is there a way to identify developing CCs without expert subjectivity? This dissertation aims to identify predictive features of developing CCs to give researchers a better understanding of TC development which will reduce the amount of deaths related to TCs. Therefore, forecasters will be able to use our research to assist in improving forecasts and preparedness for TCs which will be significant to research of weather prediction. This research incorporates difficult problems such as the complexity of CC evolution, big data, absence of ground truth data, and imbalanced data classification. These complexities are the reason this topic is identified as a difficult open-ended research area by scientists, such as Kevin E. Trenberth of the National Center for Atmospheric Research.

Determining whether a TC will develop from loosely organized CCs continues to be a difficult topic of interest (Piñeros, Ritchie, & Tyo, 2010). This is critical information when storms form close to the coast because the time to prepare and/or evacuate is short. Public officials and individual citizens alike consider this information as they plan their actions. Analysis of satellite data is an effective strategy for understanding atmospheric properties and is generally used for weather forecasting and prediction purposes (Mandal, Pal, De, & Mitra, 2005). Satellite data are used for this reason because it relates important features to physical

processes that occur in the atmosphere. One method to reliably detect or predict the development of a TC is to examine the evolution of CCs using satellite observations. CCs are too unique and complex to be incorporated into existing dynamic models since CC patterns have a variety of shapes and forms that could change rapidly (Chang, 1970; Grazzini, Bereziat, & Herlin, 2001). Due to the complexity of cloud patterns, satellite data are used to initialize these dynamic models since TCs form in areas where little or no in situ data are available (Hennon et al., 2011). Dynamic models still show discrepancies (Hennon et al., 2011); hence, it is beneficial to use only satellite data which is fully based on remote sensing of events that have actually occurred.

Identifying CCs in satellite observations is a difficult task due to the multiple definitions of a CC. Therefore, identification and tracking of individual CCs is one of the most important portions of this research since it allows to objectively analyzing the movements and identify important characteristics that contribute to the development or non-development of a CC into a TC. It is challenging to obtain enough CC cases to make valid conclusions about their complex evolution. Throughout the existence of a CC, its characteristics are obtained and then analyzed to determine what factors contribute to the formation of TCs. We still lack a complete understanding of cloud evolution and TC development. To distinguish between developing and non-developing CCs, we must thoroughly investigate CC development and how it is reflected in satellite imagery from an engineering prospective. Once this process is completed, data driven techniques provide information on CCs. This research uses data driven techniques to separate the CC data into two classes: developing and non-developing CCs. The amount of non-developing CCs outnumbers the amount of developing CCs where the imbalance ratio for the 1999-2005 North Atlantic hurricane season is approximately 27 non-developing CCs to 1

developing CC. Therefore, this problem is considered an imbalanced classification problem. To address this problem, we introduce Selective Clustering based Oversampling Technique (SCOT) which uses clustering to generate synthetic samples for the minority class (developing CCs) until the class distribution is approximately equal (C. W. Lacewell & Homaifar, 2015). Having equal class distribution is important because data imbalance is an essential source of low performance given that most classifiers assume to have balanced data. The SCOT has provided results that outscored most of the state-of-the art methods for both time series and multivariate data. Hence, we suggest it is beneficial to incorporate the SCOT into this research especially since our data is imbalanced and SCOT performs well with standard classifiers such as Classification and Regression Trees (CART), neural networks, and support vector machines (SVM).

This research will explore the development of a CC into a TC using global gridded satellite data *without* using numerical weather prediction models. Through the investigation, recommendations will be made regarding which features are predictors of TC development. Chapter 2 gives a literature review on TC development and CCs. Chapter 3 provides a literature review of feature selection techniques that can assist in the identification of the predictive features and on methods that can assist in distinguishing between developing and non-developing CCs. In this chapter, we contribute a Selective Clustering based Oversampling Technique (SCOT) which addresses data imbalance in a selective way. Chapter 4 discusses the methodology used to solve this problem such as application of thresholds, CC tracking, SCOT and sequential forward selection (SFS) as the feature selection method. Chapter 5 discusses the results of this dissertation while Chapter 6 provides case studies to further verify our techniques for identifying developing CCs. To conclude, Chapter 7 provides a summary of this dissertation and discusses some possible directions for future work.

CHAPTER 2

Literature Review: Cloud Clusters

The scientific background for TC development and CCs is presented in this chapter. A brief literature review is presented to provide basic understanding of TC formation and methods of identifying and tracking CCs.

2.1 Tropical Cyclogenesis

Tropical cyclogenesis (TCG) is a sequence of events that result in the transformation of a CC to an independent heat engine (McTaggart-Cowan et al., 2008). It is the formation of a TC whose physical processes are difficult to solve in forecast models. TCG is a rare event which only occurs in approximately 15% of the CCs in the Atlantic Ocean (Hennon, 2008). This process rarely occurs within 5^{o} of the equator due to weak Coriolis force in this region and it typically occurs over a tropical ocean (Houze, 2010; Lin, 2007). TCG typically begins with a poorly organized CC which lacks a well-defined circulation center. A large region of warm ocean water can transform a poorly organized CC into a better-defined CC because TCs are driven by the evaporation of warm water. Therefore, it is necessary to have a sea surface temperature (SST) greater than 26.5^{o}C (299.65 K) (Hennon, 2008; Houze, 2010; Terry, 2007). TCs release energy as a result of the atmosphere attempting to attain equilibrium between the warm SSTs and the cool atmosphere through convection and condensation (Terry, 2007).

Due to lack of in situ observations over tropical oceans, TCG continues to be an atmospheric phenomenon in which we lack understanding (Peng, Fu, Li, & Stevens, 2012). Many mechanisms were proposed to explain TCG, including: cooperative intensification, linear conditional instability of the second kind (CISK), wind-induced surface heat exchange (WISHE), vortex interaction, hot-tower mechanism (Lin, 2007), and the more recent pouch theory

(Montgomery et al., 2012). Multiple studies suggest TCG is dependent on CCs of various scales and their interaction with the atmosphere (Kerns & Chen, 2013). Therefore, a thorough investigation of CCs is necessary.

2.2 Identification of Cloud Clusters

To investigate the development of a TC from a CC, a precise definition of a CC must be established. Recent studies used CCs to identify synoptic-scale disturbances with embedded convective clouds, such as African easterly waves (AEWs) (Hennon et al., 2011; Hennon & Hobgood, 2003; Kerns & Chen, 2013). There are few studies on the identification of CCs because it is not a trivial process but it is subjective. Some studies suggest that mesoscale convective systems (MCSs) are organized clusters of thunderstorms with a spatial scale of 100 km (~ 1º in longitude/latitude) or more (Carvalho & Jones, 2001; Lin, 2007). Simpson et al. (1997) suggests that CCs are comprised of multiple MCSs. Hennon and Hobgood (2003) suggest that CCs are considered MCSs when they have a lifespan of at least 6 hours and a spatial scale of 250-2500 km. Based on these definitions, MCSs and CCs are used interchangeably.

Due to the scarcity of data in the North Atlantic Ocean, using satellite data to identify and track CCs is beneficial (Piñeros et al., 2010). CCs are large, long-lasting group of cumulonimbus clouds that are easy to recognize in infrared (IR) satellite images (Carvalho, Lavallée, & Jones, 2002). Forecasters rely on satellite data when in situ or direct aircraft reconnaissance is not available. CCs are usually circular in shape but smaller convective systems can sometimes merge into larger elliptical CCs. Williams and Houze (1987) suggest CC shields are 100-1000 km in dimension. Feidas and Cartalis (2005) suggest CCs in satellite images are circular with diameters of 100-500 km or elliptical with diameters up to 1000 km. Based on IR satellite imagery, Hennon and Hobgood (2003) uses the following criteria to

objectively identify potential CCs over the Atlantic basin during the 1998-2000 Atlantic hurricane seasons:

I. Each cluster must be independent and not related to a cyclone

II. A cluster cannot be elongated and must have a diameter of at least 4°

III. A cluster must be located to the south of 40°N

IV. A cluster must last for at least 24 hours

The authors use a subjective method for identifying CCs by using satellite brightness temperatures (BTs) for visual inspection instead of using an objective automated method. Using these criteria, Hennon and Hobgood (2003) label each CC as developing or non-developing. The authors classify CCs as developing if they become a tropical depression (TD) within 48 hours. Table 1 provides a summary of the characteristics of the CCs from Hennon and Hobgood (2003).

Table 1

Summary of cloud clusters from Hennon and Hobgood (2003) for the 1998-2000 Atlantic hurricane seasons

	1998	**1999**	**2000**
Total # of clusters	90	91	110
Longest in duration (hours)*	198	258	294
Mean duration (hours)*	58.9	55.1	54.8
Median duration (hours)*	42	36	42
Number of TDs	14	16	18

* *Non-developing CCs only*

The reason CCs are identified easily in IR satellite imagery is due to the temperature difference between the cold cloud tops and the warmer surface, and low cloud temperatures

(Mapes & Houze, 1993). Therefore, using a BT and a minimum area threshold offers a better understanding of the spatial and temporal characteristics (Boer & Ramanathan, 1997; Futyan & Del Genio, 2007; Vila, Machado, Laurent, & Velasco, 2008). A CC should have the potential to develop into a TC; therefore, the CC must have ample size, it must endure for an extended period of time, and the possibility of TCG must exist in the region of the CC (Hennon & Hobgood, 2003). The size and time requirements vary throughout different studies. Table 2 displays various BT and area thresholds used by numerous studies. The varieties in the BT and area thresholds demonstrate the vagueness of the many definitions of a CC. Hennon et al. (2011) identifies North Atlantic CCs as CCs that do not occur over land, covers approximately 90% of a 1° radius circle (34,800 km^2), and whose pixels have a BT less than or equal to 224 K (-49.15°C).

Table 2

Brightness temperature and area thresholds used in references

BT Threshold	Area Threshold	Region	Reference
208 K	$Diameter \geq 80\ km$	Western North Pacific	(Kerns & Chen, 2013)
213 K	$Area \geq 5000\ km^2$	Maritime continent	(Williams & Houze, 1987)
223 K		Greek peninsula	(Feidas & Cartalis, 2005)
224 K	$Radius \geq 111\ km$ $Area \geq 34{,}800\ km^2$	North Atlantic	(Hennon et al., 2011)
232 K 244 K 254 K	$Radius > 300\ km$	Atlantic and Africa	(Futyan & Del Genio, 2007)
233.15 K		Africa	(Arnaud, Desbois, & Maizi, 1992)
235 K	$Radius \geq 100\ km$	South America	(Carvalho & Jones, 2001)
235 K	$Area \geq 2{,}400\ km^2$	South America	(Vila et al., 2008)
235 K		Western Pacific	(Sherwood & Wahrlich, 1999)
240 K	$Radius > 100{,}000\ km$	Western and central Pacific	(Boer & Ramanathan, 1997)
241 K	$Area \geq 30{,}000\ km^2$	China	(Guo, Dai, & Wu, 2008)
245 K		Americas	(Machado, Rossow, Guedes, & Walker, 1998)

Boer and Ramanathan (1997) introduce detect and spread (DAS) cloud identification method which identifies clouds using multiple thresholds instead of one. Initially, CCs with BTs colder than 240 K (-33.15ºC) are identified as individual CCs. The authors spread the CC by using a new threshold which is 20 K warmer than the detecting threshold. This process is repeated for multiple detecting thresholds of 255 K (-18.15ºC), 270 K (-3.15ºC), and 285 K (11.85ºC). Futyan and Genio (2007) use the DAS method in their study of deep convective system evolution over the Atlantic Ocean and Africa. To identify cold core systems, the authors use an initial threshold of 232 K (-41.15ºC). These cold core systems are spread until a 244 K (-29.15ºC) threshold is met. In their study, this second threshold is of importance. If multiple cold core systems lie in a single region of warmer cloud, this indicates that the cold core systems share the warmer anvil cloud. On the other hand, if a new cloud region surfaces under the second threshold and does not contain a cold core system, the new cloud region is considered a new system. These warmer systems are spread to a 254 K (-19.15ºC) threshold which determines the spatial extent of the cloud. Futyan and Genio (2007) suggest that using the DAS method allows easier tracking of CCs through development stages where a cold core is not present. Therefore, it provides more information on the spatial and temporal structure of a CC than a single threshold method can provide. There are a variety of thresholds and parameters used to identify CCs in satellite imagery. Generally, a radius of at least 100 km and BTs below a threshold of 245 K (-28.15ºC) can identify CCs satisfactorily because it usually confirms the presence of deep convection (Carvalho & Jones, 2001; Vila et al., 2008).

2.3 Cloud Cluster Feature Extraction

There are few large scale factors that are favorable for TCG. These factors include having an instable atmosphere, a moist mid-troposphere, a warm ocean, near-zero vertical shear,

a large region of preexisting convection, and adequate planetary vorticity which indicates that the CC must be at least 5º latitude from the equator (Gray, 1968). There are statistical distinctions between the environment of developing and non-developing CCs but the large scale factors cannot distinguish these differences entirely (Kerns & Chen, 2013). Some of the large scale factors are difficult to determine from satellite data and atmospheric analysis products; however, it is essential to determine additional factors to distinguish between development and non-development of CCs.

Satellite data enables researchers to examine patterns in actual events. Piñeros et al. (2010) proposes an objective technique to distinguish between developing and non-developing CCs during TCG. In this study, satellite data are used because of their consistency in detecting and predicting CC evolution. The authors conclude that the underlying TC vortex structure helped symmetrically organize the CCs. Since vortices are characterized by high levels of organization, their detection at early stages of the lifecycle of TCs makes it possible to determine when CCs develop. In addition, this technique shows potential to discriminate non-developing from developing CCs.

Hennon and Hobgood (2003) hypothesized that certain features are predictors of TCG. The authors recommend that the most significant predictor is the daily genesis potential which is the difference between the 900 hPa and 200 hPa relative vorticities. This predictor is calculated using reanalysis data from the National Centers for Environmental Prediction-National Center for Atmospheric Research (NCEP-NCAR). The authors convey that this feature is of importance because TCG requires near-zero vertical wind shear near the center of the storm and a vertical shear gradient that is strong. Therefore, a more favorable development environment is obtained when the daily genesis potential is high.

It is suggested by multiple studies that for TCG to occur, the CC must be at least 5° latitude away from the equator. Therefore, the next most significant predictor in Hennon and Hobgood (2003) was a Scaled Coriolis (SC) parameter which is defined as follows:

$$f = 2\omega \times 10^4 \sin(\phi)$$

where ϕ denotes the latitude in degrees and ω indicates the angular rotation of the Earth ($\omega = 7.29 \times 10^{-5}\ s^{-1}$). Hennon and Hobgood (2003) list other predictors, such as maximum potential intensity and precipitable water, but they are much weaker in significance. When using solely IR gridded satellite observations, daily genesis potential is impossible to calculate; therefore, for this research, only the scaled Coriolis parameter is relevant.

2.4 Tracking Cloud Clusters

The movement of clusters provides important information about CC evolution. Chang (1970) suggests that a longitude-time (Hovmöller) diagram from zonal strips of successive satellite images provides useful information since most wave motions propagate zonally. This is important because some studies suggest that wave motions, such as the AEWs, can initiate TC formation in the Atlantic basin (Berry & Thorncroft, 2005; Hopsch, Thorncroft, & Tyle, 2010; C. Lacewell, Homaifar, & Lin, 2013; Lin, 2007; Lin, Liu, Tang, Spinks, & Jones, 2013; Peng et al., 2012; Reed, Norquist, & Recker, 1977). The Hovmöller diagram is not a dependable method because it is a very subjective way to track the MCSs or the AEWs.

Arnaud et al. (1992) uses an automatic tracking technique to track convective systems that propagate from West Africa to the Atlantic Ocean. In this method, a cloud is tracked based on the intersection between the clouds in two successive images. If more than one cloud intersects, the cloud with the maximum intersection is the tracked cloud as long as at least half of its area intersects. Similarly, Kerns and Chen (2013) tracks CCs by viewing hourly consecutive

satellite images. In these images, a CC is considered the same CC if at least 50% or 10,000 km^2 overlap between the images.

Carvalho and Jones (2001) proposes the maximum spatial correlation tracking technique (MASCOTTE) which uses the BTs from Geostationary Operational Environmental Satellite (GOES-8) images to automatically characterize and track convective system (CS) properties. In MASCOTTE, CSs are identified as systems with a radius of at least 100 km and BTs less than or equal to 235 K. At time *t*, an individual CS is identified and the CS with the maximum spatial correlation at time *t+1* is considered the new location of that CS. In this technique, splitting is identified when multiple CSs at time *t+1* have positive and high spatial correlation with the CS at time *t*. On the other hand, merging is identified when the area increases and the spatial correlation decreases for more than 10% but remains positive.

Mandal et al. (2005) proposes a novel hierarchical method to find tracer clouds from satellite images using cloud motion vectors to study the dynamic behavior of clouds. This method extracts features from a sequence of cloud images and uses them to calculate several parameters such as mean, standard deviation, and entropy. Based on these features, tracer clouds are identified and cloud motion vectors are used to make predictions on storm movement.

Boer and Ramanathan (1997) developed an automatic cloud tracking algorithm (CTA) to track CCs. In the CTA, individual CCs are identified by DAS and are replaced with their equivalent ellipse which has the same characteristics as the actual CC, i.e. centroid, eccentricity, area, and orientation. In the tracking method, two CCs in two successive images are considered the same CC if either centroid falls inside the overlap of the ellipses. The CC undergoes splitting if multiple CCs at time *t+1* overlap with a single CC at time *t*. The CC merges if multiple CCs

at time *t* overlap with a single CC at time *t+1*. Please see Appendix A for details regarding features such as centroid and eccentricity.

Vila et al. (2008) introduces Forecasting and Tracking the Evolution of Cloud Clusters (ForTraCC) to track and forecast CC properties using satellite images. In ForTraCC, the CCs are identified using BT and area thresholds. The tracking method used is based on an area overlap method. To track CCs, each CC is given a CC number for each time step and one of the five conditions could occur:

1. *Spontaneous generation:* This occurs when a new CC generates. In this case, there is no CC visible in satellite image at time *t* but a new CC is visible at time *t+1*.
2. *Natural dissipation:* This occurs when a CC dissipates. In this case, there is a CC present at time *t* but not at time *t+1*.
3. *Continuity:* This occurs when there is an overlap in two successive images between only one pair of CCs.
4. *Split:* This occurs when one CC at time *t* overlaps with multiple CCs at time *t+1*. In this case, the continuing CC is considered as the CC with the maximum overlapping area and the other overlapping CCs at time *t+1* are considered new CCs.
5. *Merger:* This occurs when one CC at time *t+1* overlaps with multiple CCs at time *t*. In this case, the continuing CC is considered as the CC with the maximum overlapping area and the other overlapping CCs at time *t* are considered CCs that have dissipated.

A detecting and tracking algorithm for CCs is proposed by Hennon et al. (2011). A dataset comprised of all tropical CCs over water from 1980 to 2008 is created using the algorithm. This algorithm uses Gridded Satellite (GridSat) and International Best Track Archive for Climate Stewardship (IBTrACS) data to produce the Tropical Cloud Cluster (TCC) dataset. The authors use the geometric center to track the identified CCs through time intervals. The authors use a tracking framework similar to an area overlap method with a search range incorporated. In this tracking method a CC is considered the same CC in the next time step if it is within a specified distance. Due to the fact that CCs may disappear for up to 12 hours, Hennon et al. (2011) uses a search radius for up to 12 hours with 3 hour increments. In cases where the CC does disappear, all coordinates are estimated through a linear interpolation between the last known coordinates and all other CC features are labeled "missing." After tracking all identified CCs, all CCs that did not last for at least 24 hours, with the exception of CCs that developed into TCs, were removed. This dataset makes thousands of cases of TCG immediately available to researchers. These researchers can focus on identifying large-scale differences between developing and non-developing CCs. Their algorithm excludes all CCs which are located over land and it does not consider any other factors which may contribute to a CC's development.

Our prior work, documented in Lacewell et al. (2013), uses the Scale and Orientation Adaptive Mean Shift Tracking (SOAMST) method to track pre-TS Debby (2006) to its origin in eastern North Africa. This method solves problems in estimating the scale and orientation changes in objects; therefore, it has been helpful in tracing processes such as cloud movement.

2.5 Summary

This chapter presented a review of TCG, and identification, features extraction, and tracking of CCs. These topics are pertinent to understanding the background information needed to distinguish between the development and non-development of CC. Discussions of other relevant topics of this research are in the succeeding chapters.

CHAPTER 3

Literature Review: Feature Selection and Classification of Imbalanced Data

The scientific background of feature selection and addressing a two-class classification problem using imbalanced data is presented in this chapter. A brief literature review is offered to provide basic understanding of feature selection techniques, balancing imbalanced data, and performance measures to assess the classification of imbalanced data.

3.1 Feature Selection Techniques

A significant challenge in machine learning is the high dimensionality of data. These datasets may contain redundant features which may reduce classification performance, have high computation costs, and in our case, poor identification of developing CCs (Brown, Pocock, Zhao, & Luján, 2012; Y. Chen, Li, Cheng, & Guo, 2006; Song, Ni, & Wang, 2013). An exhaustive evaluation of feature subsets in such datasets are unfeasible because, in this case, it would involve the evaluation of $\frac{80!}{M!(80-M)!}$ combinations if we choose to reduce the eighty features to M features (Wilder, 2011). To address this issue, this section provides necessary background on feature selection techniques.

Feature selection is a pre-processing step for high dimensional data used to alleviate the *curse of dimensionality* by reducing the number of features, storage, and computation time in statistical learning such as classification (Y. Chen et al., 2006; M. Han & Liu, 2013; MathWorks Incorporated, 2014). Feature selection is necessary in identifying the predictive features of developing CCs since eighty features are extracted from each CC in the dataset that contains thousands of observations. To identify the predictive features, we must discover a robust subset of features that can satisfactorily distinguish between developing and non-developing CCs.

There are three categories of feature selection: filters, wrappers, and embedded methods. Filter methods use attributes of the data to assess and select a subset of generic features with only few assumptions. Therefore, they are classifier independent techniques. On the other hand, wrapper and embedded methods are classifier dependent. Wrapper methods use the performance of a pre-selected classifier to search for feature subsets. This is a benefit in generalization but a drawback can occur in computational cost and it can become specific to the chosen classifier. Embedded methods perform feature selection in the training process and assume precise model structure (Brown et al., 2012; Y. Chen et al., 2006; Dias, Kamrunnahar, Mendes, Schiff, & Correia, 2010; Pohjalainen, Räsänen, & Kadioglu, 2013; Saeys, Inza, & Larrañaga, 2007; Song et al., 2013). This dissertation focuses on subset selection and scoring algorithms to assist in identifying a subset of features as predictive features since they are typically simple to implement. For the remainder of this section, we use a standard notation to represent the data and features. We consider a set of m observations containing n features and a set of class labels denoted by $\{CC_j^i, Class_j\}$ where CC_j^i is the j[th] observation of the i[th] feature for $j = 1, 2, \dots, m$ and $i = 1, 2, \dots, n$.

3.1.1 Feature subset selection algorithms. Feature subset selection algorithms are feature selection techniques that identify a subset of features from a given dataset while removing irrelevant and redundant features (Yoon, Yang, & Shahabi, 2005). The following feature subset selection algorithms are included in this section: sequential forward/backward selection and random subset feature selection.

Two well-known and widely used feature selection techniques are the sequential forward selection (SFS) algorithm which is proposed by Whitney (1971) and the sequential backward selection (SBS) algorithm which is originally described in Marill and Green (1963). These

feature selection methods are wrapper methods. The SFS algorithm selects a subset of features by beginning with an empty set of features and sequentially adding features until there is no change in the performance of the pre-selected classifier. On the other hand, the SBS algorithm begins with a set of all features and sequentially removes features until there is no change in the performance of the pre-selected classifier (Blachnik, 2009; Marill & Green, 1963; Pohjalainen et al., 2013; Whitney, 1971). In the iterations of SFS, the feature that is added to the subset maximizes the selected classification performance measure c_p. In SBS, the feature that mostly affects the classification performance is removed. Each feature is assessed individually therefore SFS and SBS can be computationally expensive which is dependent on the total number of features (Dias et al., 2010; MathWorks Incorporated, 2014; Pohjalainen et al., 2013; Wilder, 2011).

Räsänen and Pohjalainen (2013) proposes the Random Subset Feature Selection (RSFS) wrapper method that attempts to discover a subset of features by repetitively choosing a random subset of features and comparing its classification results (Pohjalainen et al., 2013; Räsänen & Pohjalainen, 2013). In RSFS, there are true features f_t with associated relevance value $r_t \in [-\infty, \infty]$ and dummy features d_y with associated relevance value g_y. During each iteration k, the following steps of RSFS are performed:

1. A subset S_k containing p features is randomly selected from n features using a uniform distribution.
2. Use k nearest neighbor (NN) classification on the data using S_k and compute a desired classification performance measure c_k.
3. Update relevance value r_t of the true features f_t using

$$r_t \leftarrow r_t + c_k - E\{c\}$$

where $E\{\cdot\}$ is the expected value. In parallel, update relevance value g_y of the dummy features d_y using

$$g_y \leftarrow g_y + c_k - E\{c\}$$

which essentially becomes a random walk process and provides a baseline level r_{rand}. A true feature must exceed this baseline level to become an important feature.

4. Return to Step 1 with a new random subset.

To identify the best subset of features, r_t must satisfy

$$p(r_t > r_{rand}) \geq \delta \; \forall \; f_t$$

where δ is a user-defined threshold for probability. Räsänen and Pohjalainen (2013) set the probability threshold to 0.99. The cumulative normal distribution is used to obtain the probability that f_t is more relevant than g_y using

$$p(r_t > r_{rand}) = \frac{1}{\sigma_g \sqrt{2\pi}} \int_{-\infty}^{r_t} exp\left(\frac{-\left(CC^i - \mu_g\right)^2}{2\sigma_g^2}\right) dx$$

where μ_g and σ_g denote the mean and standard deviation of the relevance values of all dummy features. Refer to Pohjalainen et al. (2013) and Räsänen and Pohjalainen (2013) for further details.

3.1.2 Scoring Algorithms. The fastest and simplest types of filter methods in feature selection problems are the scoring algorithms which are also called ranking methods. These methods use a scoring function computed from CC_j^i and $Class_j$ to identify valuable features. These methods only involve the computation of n scores which are ranked by their significance according to the given score (Pohjalainen et al., 2013). Additional considerations are needed to determine the size of the feature subset when using the calculated scores. This is an additional

issue of using these methods but it is not discussed in this dissertation. The following scoring algorithms are included in this section: statistical dependency, box difference index, independent significance features test, and correlation ranking.

Statistical dependency (SD) determines whether the feature values are dependent on the class labels. In this method, each feature value is quantized in a manner where an equal amount of samples is contained in each bin when quantizing the entire dataset. The following equation is used to calculate the statistical dependence between the discretized feature value f_j and $Class_j$

$$SD^i = \sum p(f_j, Class_j) \frac{p(f_j, Class_j)}{p(f_j)p(Class_j)}$$

where larger SD values indicate a higher dependency between f_j and $Class_j$. SD of the minimal value 1 indicates the feature is fully independent of the class labels (Pohjalainen et al., 2013).

Peng et al. (2012) proposes a box difference index (BDI) to objectively and quantitatively identify predictive parameters of a tropical disturbance developing into a TC. The BDI is defined as follows:

$$BDI = \left| \frac{M_{Dev} - M_{Nondev}}{\sigma_{Dev} + \sigma_{Nondev}} \right|$$

where M_{Dev} and σ_{Dev} (M_{Nondev} and σ_{Nondev}) represents the mean and standard deviation of the considered feature for developing (non-developing) CCs. Higher BDI magnitudes represent a better predictability of the variable. The BDI is used on many key genesis parameters for the North Atlantic basin and the authors conclude that a parameter with larger BDI amplitude contributes more to the prediction of TCG than one with a lesser amplitude. Figure 1 illustrates an example from Peng et al. (2012) of the BDI for special cases of relative humidity. When the BDI value equals zero, the developing and non-developing CCs are similar. When the BDI value equals 0.5, the two groups are partially separated but when the BDI equals one, the

developing and non-developing CCs are well separated. By using this index, the authors recommended that thermodynamic parameters, i.e. SST, are more important than dynamic parameters, i.e. vertical shear, when distinguishing between CCs in the North Atlantic Ocean.

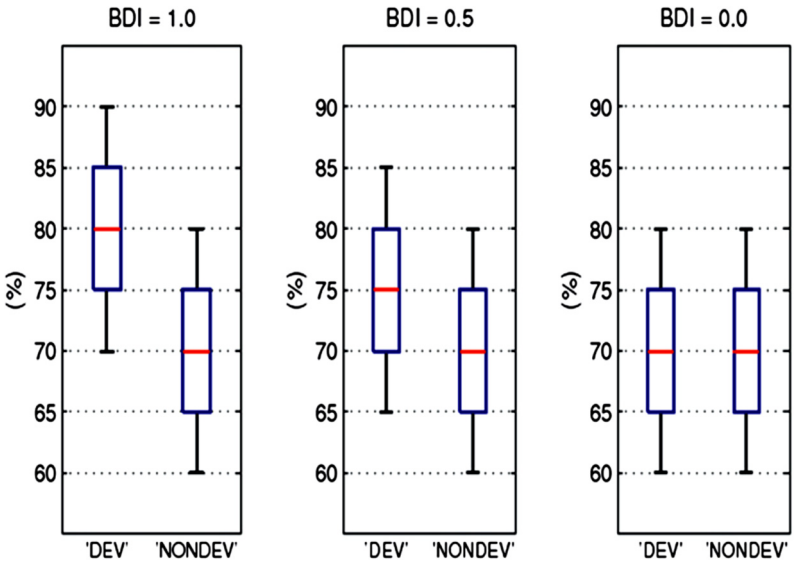


Figure 1. Box-and-whiskers figure from Peng et al. (2012) where the box difference index varies for special cases of relative humidity.

Weiss and Indurkhya (1997) propose the independent significance features test which is also called Fisher's discriminant ratio (FDR). This filter method measures the linear discriminating power of features using the following equation:

$$FDR^i = \frac{\left(\mu_{DV}^i - \mu_{ND}^i\right)^2}{\left(\sigma_{DV}^i\right)^2 + \left(\sigma_{ND}^i\right)^2}$$

where μ_{DV}^i and μ_{ND}^i denote the means of developing and non-developing CCs of feature i and $\left(\sigma_{DV}^i\right)^2 + \left(\sigma_{ND}^i\right)^2$ represent the within-class variances of the data. Better features have higher FDR values.

The correlation ranking is a filter method that can only discover linear dependencies between a feature and the associated class labels. This method uses the Pearson correlation coefficient to score each feature which is defined as

$$R(i) = \frac{cov(CC^i, Class)}{\sqrt{var(CC^i) * var(Class)}}$$

where CC^i is the i[th] feature, $cov(\cdot)$ is the covariance, and $var(\cdot)$ is the variance (Chandrashekar & Sahin, 2014; Guyon & Elisseeff, 2003). If the inputs are not random variables, the estimate is given by

$$R(i) = \frac{\sum_{j=1}^{m}(CC_j^i - \overline{CC}^i)(Class_j - \overline{Class})}{\sqrt{\sum_{j=1}^{m}(CC_j^i - \overline{CC}^i)^2 \sum_{j=1}^{m}(Class_j - \overline{Class})^2}}$$

where the bar notation indicates the average over j (Guyon & Elisseeff, 2003).

3.2 Techniques for Imbalanced Data

In most real world applications, there is a demand to accurately identify rare events which are typically more significant than frequently occurring events (Bekkar & Alitouche, 2013; Fernández, García, & Herrera, 2011; He, Bai, Garcia, & Li, 2008; G. M. Weiss, 2013). In these cases, the observed data are highly imbalanced which causes a decrease in classification accuracies due to standard classifiers that assume the class distribution of data are approximately equal (Batista, Prati, & Monard, 2004; Cao, Li, Woon, & Ng, 2013; He & Garcia, 2009). When the class distribution is not equal, the classifiers perform better on the majority (larger) class than the minority (smaller) class. The identification of predictive features of CCs which will develop into a TC is considered an imbalanced data problem because the number of non-developing CCs is expected to outnumber developing CCs. In this application, classifying a developing CC

accurately is of great importance and misclassifying a developing CC is more costly than misclassifying a non-developing CC.

To address the imbalanced learning problem, many approaches have been introduced. Sampling methods, cost based methods, kernel based methods, and active learning methods are four categories of imbalanced learning solutions (Barua, Islam, Yao, & Murase, 2014; He & Garcia, 2009). The sampling methods are the focus of this dissertation because this category performs at the data level where the class distribution is modified and the techniques can be used with standard classifiers. Data level approaches generally provide better results than algorithmic level approaches which are more data specific (Barua et al., 2014; Cao et al., 2013; He & Garcia, 2009)

In recent years, sampling methods have been used successfully to modify the class distribution of imbalanced data (Barua et al., 2014; Chawla, Bowyer, Hall, & Kegelmeyer, 2002; H. Han, Wang, & Mao, 2005; He & Garcia, 2009). These methods balance the amount of samples in each class by either reducing the majority class samples (undersampling), increasing the minority class samples (oversampling), or by combining the two methods. Hence, any classifier can use sampled data instead of modifying the classifier to fit the data.

3.2.1 Undersampling techniques. Random sampling is the most simplistic, non-heuristic type of sampling. Random undersampling removes instances from the majority class randomly until the class distributions are approximately equal (Batista et al., 2004; Bekkar & Alitouche, 2013; García, Sánchez, Mollineda, Alejo, & Sotoca, 2007; Japkowicz, 2000). This method is less frequently used in classification problems because it may remove valuable information from the majority class (Batista et al., 2004). Many undersampling techniques have been introduced to assist in improving the classification performance of imbalanced data (Batista et al., 2004;

Bekkar & Alitouche, 2013; Hart, 1968; Kubat & Matwin, 1997; Wilson, 1972). A Tomek link is a pair of NNs of opposite classes, which are minimally distanced. When Tomek links are identified, either both samples are on the decision boundary or one of the samples is noise. When this method is used as an undersampling technique, only the sample in the link belonging to the majority class is removed (Batista et al., 2004; Bekkar & Alitouche, 2013).

Hart (1968) introduces the condensed nearest neighbor (CNN) rule which was based on the NN rule. In the CNN rule, it identifies a consistent subset of samples, which classifies the remaining samples correctly. To determine the consistent subset, the CNN rule initializes a subset called *STORE* with one randomly selected majority sample and all minority samples in the dataset. The remaining samples are classified using the NN rule using the contents in *STORE* as a reference set. If the samples are classified correctly, they are added to a subset called *GRABBAG*; otherwise, the sample is placed in *STORE*. This method continues to loop through *GRABBAG* until all samples are transferred to *STORE* or until the algorithm loops through one complete pass through *GRABBAG* without any additional transfers. The contents of *STORE* are considered the consistent subset and are used as reference samples for the NN rule. The contents of *GRABBAG* are the majority samples that are distant from the decision boundary which are removed from the dataset (Batista et al., 2004; Hart, 1968).

Kubat and Matwin (1997) introduce an undersampling method called one-sided selection (OSS) which combines both Tomek links and the CNN rule. In this method, Tomek links are used to remove noisy and borderline majority samples since these samples are easily misclassified. After these samples are eliminated, the CNN rule is applied to remove majority samples that are distant from the decision boundary (Batista et al., 2004). Batista et al. (2004) introduces CNN plus Tomek links which is an undersampling method similar to OSS except that

the algorithms are performed in reverse order. The author suggests that this method is competitive with OSS and may be less computationally expensive since the Tomek links are identified on a reduced dataset.

Wilson (1972) proposes the edited nearest neighbor (ENN) rule which identifies the three NNs of all observations and removes samples whose class label differs from at least two of its three NNs (Batista et al., 2004). Laurikkala (2001) proposes Neighborhood Cleaning Rule (NCL) as an undersampling technique which incorporates ENN. In this method, the ENN rule is applied to all samples, x_i. If x_i is a member of the minority class and almost all of its NNs are from the majority class, then the NNs belonging to the majority class are removed. On the other hand, x_i is removed when it is a member of the majority class and two of its three NNs are from the minority class.

3.2.2 Oversampling techniques. Random oversampling duplicates randomly selected minority class samples until the class distributions are nearly equal (Batista et al., 2004; Bekkar & Alitouche, 2013; García et al., 2007; Japkowicz, 2000). This method can lead to overfitting the minority class since it replicates existing samples (Batista et al., 2004).

Another oversampling method is synthetic oversampling. Synthetic oversampling generates synthetic minority class samples to assist in improving the classification performance of imbalanced data (Barua et al., 2014; He & Garcia, 2009). Chawla et al. (2002) introduces Synthetic Minority Oversampling TEchnique (SMOTE) which operates in the feature space to overcome the overfitting short-coming of random oversampling (Chawla, 2005; Chawla et al., 2002; García et al., 2007; He & Garcia, 2009). It is an effective method and it is basis of many other synthetic oversampling techniques. In SMOTE, synthetic data are constructed for each minority sample until the class distribution is approximately equal. The synthetic samples are

formulated along a line segment that joins the selected minority sample and one of its randomly selected k NNs (Barua et al., 2014; Bekkar & Alitouche, 2013; Chawla et al., 2002; H. Han et al., 2005; Luengo, Fernández, García, & Herrera, 2011). The number of randomly selected k NNs chosen is dependent on the amount of oversampling needed. Chawla et al. (2002) set k to five. If the number of minority samples should be doubled to make the class distribution approximately equal (200% of the original minority class), then two of the five NNs are selected and each NN is used to generate a synthetic sample. The following equation is used to create a synthetic minority sample, s:

$$s = x + \delta(y - x) \tag{1}$$

where x is the minority sample, y is the randomly selected k-NN of x, and $\delta \epsilon [0, 1]$ is a random number. Eq. (1) interpolates between similar minority samples instead of duplicating the samples. Hence, the overfitting problem of random oversampling is addressed (Luengo et al., 2011). In contrast, it can cause over generalization which can produce more overlapping between classes (Barua et al., 2014; Batista et al., 2004; He & Garcia, 2009). The time complexity of SMOTE for the worst situation is $O(|x| \times v \times \gamma)$ where $|x|$ denotes the number of minority samples, v represents the number of synthetic samples generated for each minority samples, and γ denotes the cost of calculating the k nearest neighbor. The value of γ depends on the approach taken. For example, $\gamma = k|x|$ if the exhaustive search algorithm is used for finding the nearest neighbor; therefore, the time complexity is $O(|x|^2 \times v)$. Figure 2 uses the well-known Fisher's iris data, which consists of 150 iris samples with the sepal length, the sepal width, the petal length, and the petal width measurements as features. To demonstrate the generation of synthetic samples and the capability of the selected oversampling methods, only the sepal length and sepal width features are used. The minority class contains the setosa species

and the majority class contains the versicolor and virginica species. The sample distribution of the data when using SMOTE for oversampling is displayed in Figure 2 which is from Lacewell and Homaifar (2015). In the figure, there are synthetic samples that are replicas of the minority class samples.

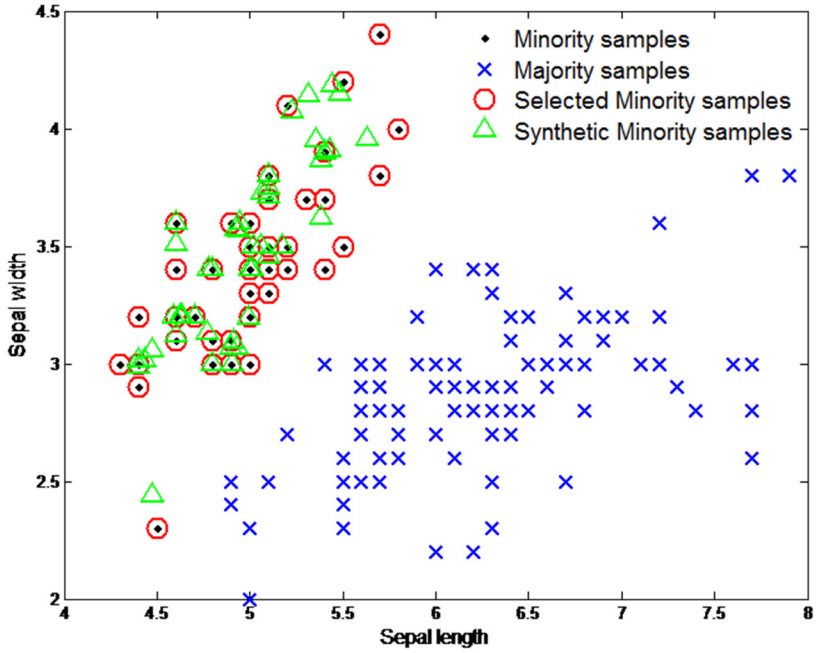


Figure 2. Sample distribution of Fisher's Iris data when using Synthetic Minority Oversampling TEchnique (C. W. Lacewell & Homaifar, 2015).

Han et al. (2005) introduces a modification of SMOTE named Borderline SMOTE. The difference between SMOTE and Borderline SMOTE is that the latter creates synthetic samples for minority samples that lie closer to the decision boundary because these samples are difficult to learn by a classifier. A minority sample where over half of its *k*-NNs are members of the majority class is considered a borderline minority sample. In this method, *k* is user defined. Borderline SMOTE does not generate samples for noisy minority samples, which are minority samples whose NNs all belong to the majority class. Instead it uses Eq. (1) to create synthetic

samples for the borderline minority samples (Barua et al., 2014; H. Han et al., 2005; He & Garcia, 2009). The time complexity of this method is $O(|x| \times |x_{border}| \times v)$ where $|x_{border}|$ denotes the number of borderline minority samples. In this case, the k nearest neighbor algorithm uses the exhaustive search approach. Borderline SMOTE typically performs better than SMOTE since it concentrates on minority samples with higher chances of being misclassified (Bekkar & Alitouche, 2013). Figure 3, from Lacewell and Homaifar (2015), displays samples of the Fisher's iris data once Borderline SMOTE is applied. As displayed in the figure, in some cases, Borderline SMOTE does not recognize samples closer to the decision boundary as borderline samples because its k NNs are from the minority class. This drawback can be a challenge in classification because of insufficient information regarding the minority class samples near the decision boundary.

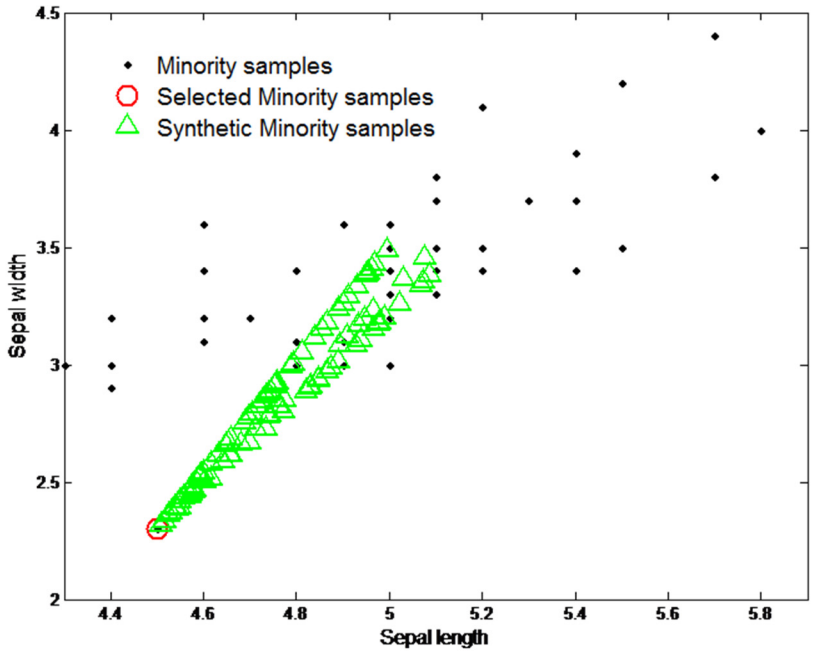


Figure 3. Synthetic samples generated by Borderline Synthetic Minority Oversampling TEchnique on Fisher's Iris data (C. W. Lacewell & Homaifar, 2015).

Hu et al. (2009) proposes Modified Synthetic Minority Oversampling Technique (MSMOTE) which improves SMOTE by considering the distribution of minority samples and noise. In this method, the minority samples are divided into three groups: security samples, border samples and noise samples. Security samples are minority samples whose *k* NNs are members of the minority class. Noise samples are minority samples whose *k* NNs are members of the majority class. Border samples are minority samples, which are neither security samples nor noise samples. MSMOTE generates synthetic samples for security and border samples but it does not generate samples for the noise samples. As in SMOTE, synthetic samples are generated using Eq. (1) where $x \in \left(S_{security} \cup S_{border}\right)$. If $x \in S_{security}$ then y is a randomly selected *k* NN of x. On the other hand, if $x \in S_{border}$ then y is the NN of x.

He et al. (2008) proposes an adaptive synthetic (ADASYN) sampling approach that generates samples for hard to learn minority samples. Unlike SMOTE or Borderline SMOTE, ADASYN uses a density distribution, $\hat{r_i}$, to determine the number of synthetic minority samples to generate for each minority sample. For each minority sample, k NNs are identified. The density distribution is calculated as

$$\hat{r_i} = \frac{\frac{\Delta_i}{k}}{\sum_{i=1}^{m_s} \frac{\Delta_i}{k}}$$

where Δ_i denotes the number of majority samples in the k NNs of the minority sample and m_s denotes the number of minority samples. The number of synthetic minority samples that should be generated for each minority sample is defined by

$$g_i = \hat{r_i} \times \beta(m_l - m_s)$$

where m_l denotes the number of majority samples and $\beta \in [0, 1]$ specifies the desired balance level after generation. The synthetic minority samples are generated in the same manner as

SMOTE using Eq. (1). The time complexity of ADASYN is equal to that of SMOTE since the only difference between the two are that ADASYN does not use uniform distribution to determine the number of synthetic samples to generate.

Barua et al. (2014) proposes a cluster based oversampling technique that is a variation of SMOTE called Majority Weighted Minority Oversampling TEchnique (MWMOTE). MWMOTE attempts to improve the drawbacks of the preceding methods. As with Borderline SMOTE and MSMOTE, MWMOTE does not generate synthetic samples for noisy minority samples. Instead of using borderline minority samples to generate the synthetic data, MWMOTE uses informative minority samples. Informative minority samples are the NNs of borderline majority samples, which are majority samples that are NNs of non-noisy minority samples. MWMOTE uses hierarchical average-linkage agglomerative clustering to assign selection weights to the minority samples in hopes to improve the synthetic sample generation process. Agglomerative clustering technique begins with each sample being a single cluster and at each level it merges clusters together based on the smallest intergroup dissimilarity until only one cluster is left at the top (Hastie, Tibshirani, & Friedman, 2009). Average linkage agglomerative clustering calculates the distance between clusters as the distance between the averages of the cluster members (Hastie et al., 2009). The selection weight for each minority sample is based on the summation of a closeness factor, C_f, multiplied by a density factor, D_f, of all borderline majority samples as defined by

$$S_w(x) = \sum_{y_i \in S_{borderline_{maj}}} C_f(y_i, x) D_f(y_i, x)$$

Sparse clusters of minority samples and samples closer to the decision boundary are assigned a higher selection weight. Barua et al. (2014) provides a more detailed description of the closeness

and density factors. MWMOTE also uses Eq. (1) to generate the synthetic minority samples where y is a member of x's cluster instead of being the randomly selected k NN of x. This minor change in Eq. (1) keeps the generated synthetic minority sample from falling in the majority class region (Barua et al., 2014). The simplified time complexity is $O(|x|^2 + |x_{imin}| \times v)$ where $|x_{imin}|$ denotes the number of informative minority samples. Figure 4 from Lacewell and Homaifar (2015) shows the synthetic samples generated by MWMOTE on the Fisher's iris data.

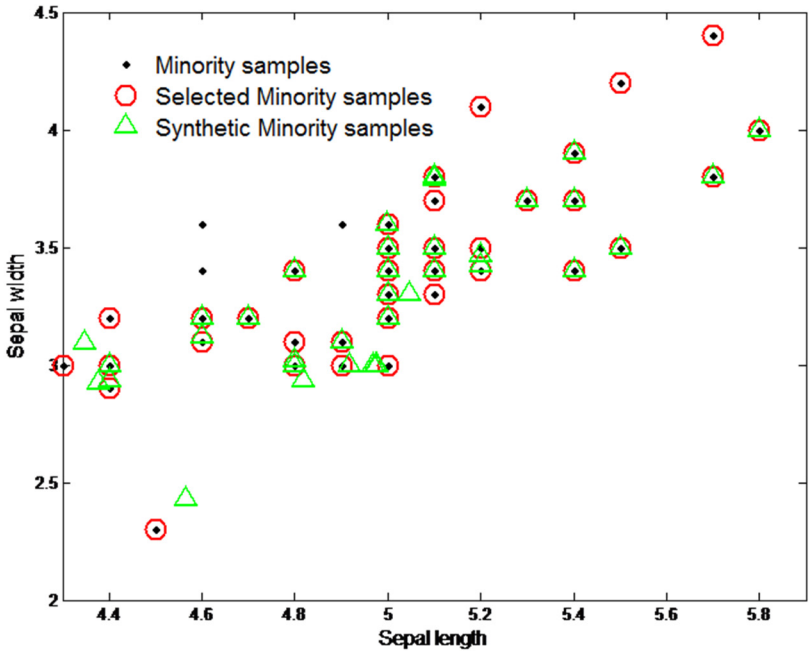


Figure 4. Synthetic samples generated by Majority Weighted Minority Oversampling TEchnique on Fisher's Iris data (C. W. Lacewell & Homaifar, 2015).

Lacewell and Homaifar (2015) propose the Selective Clustering based Oversampling Technique (SCOT) which uses a combination of the local outlier factor (LOF) to identify outliers, agglomerative clustering which best fits the data, and it explores the neighborhood of the informative minority samples to reduce the risk of overfitting when generating synthetic

samples. The LOF is a degree of objects being outliers which was introduced by Breunig et al. (2000). This degree provides a numerical representation of how isolated a sample is when compared to its surrounding neighborhood. Further details regarding the LOF is found in Breunig et al. (2000). SCOT is separated into three main processes: identifying informative minority samples, identifying informative clusters, and finally, generating synthetic samples. The complete algorithm is summarized below:

1) Compute the *k* nearest neighbor set for each minority sample according to Euclidean or standardized Euclidean distance. The *k* is equivalent to 5% of the number of minority samples. Therefore,

$$k = 0.05|S_{min}|$$

2) Construct the noisy minority set containing minority samples where all k-nearest neighbors are majority class samples. The members of this set are removed from the original dataset.
3) Construct the filtered minority set containing minority samples where the number of minority class samples, *m*, among its *k*-nearest neighbors satisfy

$$\frac{k}{2} \leq m \leq k$$

4) Construct the danger minority set and the borderline majority set. The danger minority set contains minority samples where the number of minority class samples, *m*, among its *k*-nearest neighbors satisfy $0 < \mathrm{m} < \frac{\mathrm{k}}{2}$ and the majority samples among its *k*-nearest neighbors are contained in the borderline majority set.
5) Construct the qualified minority set as

$$\mathrm{S_{qual}} = \mathrm{S_{fmin}} \bigcup \mathrm{S_{danger}}$$

6) Find the informative minority set which contains ninety-nine percent of samples of S_{qual}. One percent of the samples with the highest LOF values are considered outliers and are eliminated.
7) If there are less than two members in the informative minority set
    a) Construct the noisy test set as

$$S_{TestNoisy} = S_{qual} \bigcup S_{Noisy}$$

    b) Add noisy samples to the danger minority set that have local outlier factors less than 0.1 quantile of all local outlier factors
    c) Repeat steps 5 and 6 using new danger minority set.
8) Determine the best agglomerative hierarchical clustering tree structure based on inconsistency coefficients and cophenetic correlation coefficients. To determine the best cluster tree, the maximum inconsistency coefficients are sorted in ascending order. Out of the top three trees with high maximum inconsistency coefficients, the tree with the highest cophenetic correlation coefficient is used to cluster the data.
9) Cluster the informative minority set using a threshold, Th_{cutoff}, to separate the data into clusters. When separating the clusters, a node and its leaves should have an inconsistency coefficient less than Th_{cutoff}, which is equivalent to the median of all inconsistency coefficients.
10) For each cluster, compute the cluster center, the number of members, and the average Euclidean distance between the informative minority samples and the cluster center.
11) M informative clusters are formed where the average Euclidean distance between the informative minority samples and the cluster center is not equal to zero, and the number of samples in each cluster is greater than one but less than the number of informative minority

samples. The clusters are denoted as $L_1, L_2, \ldots, L_M$.

12) Generate a point system which is based on the ranking of the population factor, a closeness factor, and a sparseness factor to specify the number of synthetic samples to generate per cluster, $SynPts$.

13) Do for $i = 1 \ldots M$

14) Do for $j = 1 \ldots SynPts$

a) Select a sample x at random from the members of cluster L_i.

b) Select another random sample y from the members of cluster L_i

c) Generate one synthetic minority sample, s, according to $s = x + \delta(y - x)$, where δ is a number in the range $[0.1, 0.9]$.

d) Add s to a set of all generated minority samples, S_{syn}

15) End Loop

16) End Loop

17) Add S_{syn} to the original dataset

Overall, SCOT can enhance the classification performance of the minority class. SCOT's approach performs well on truly imbalanced data that contain less than ten percent minority samples. This performance is demonstrated by a comparison with state-of-the-art techniques as found in Appendix B. Unlike other methods, SCOT eliminates user defined parameters, identifies hard to learn minority samples better, produces synthetic samples to better define the decision boundary, and generates synthetic samples in the area of the minority class to avoid overlapping of classes which, in all, lowers the risk of overfitting. The simplified time complexity is $O\left(|x|^2 + \left|x_{inform\,\epsilon\,M}\right| \times v\right)$ where $\left|x_{inform\,\epsilon\,M}\right|$ denotes the number of samples in the informative clusters. The synthetic samples generated by applying

SCOT on the Fisher's iris data is illustrated in Figure 5. This technique produces more synthetic samples in areas where there are gaps in the data or where more information is needed. Further details on this method are found in Lacewell and Homaifar (2015) and simulation results are found in Appendix B for a better representation of its performance.

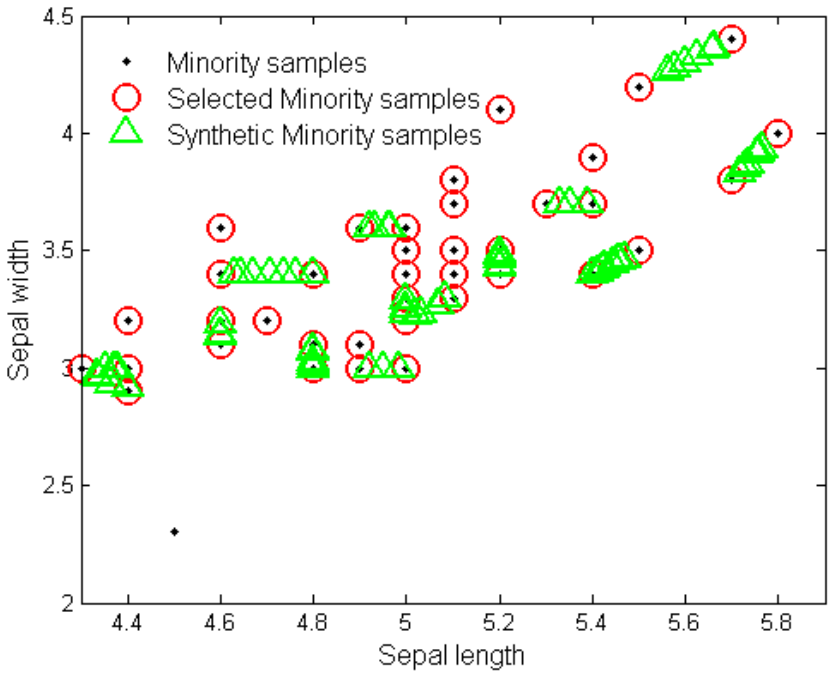


Figure 5. Synthetic samples generated by Selective Clustering based Oversampling Technique on Fisher's Iris data (C. W. Lacewell & Homaifar, 2015).

3.2.3 Data cleaning techniques. Data cleaning techniques are used to remove overlapping that may be caused by sampling techniques. Removing overlapping samples can improve classification performance by making class clusters more defined (He & Garcia, 2009). The data cleaning process removes overlapping samples which are identified using different methods. Tomek links and the ENN rule are two techniques that are used for data cleaning when combined with sampling techniques. These data cleaning methods are used to remove difficult to learn samples and are mostly applied when sampling does not provide satisfactory results.

Tomek links can either be used as an undersampling technique or as a data cleaning method. As a data cleaning method, a sampling technique is applied to the dataset to balance the class distribution followed by identifying and removing both samples of the Tomek link (Batista et al., 2004). When using ENN as a data cleaning method, it removes any sample where the class of majority of its nearest neighbors differ from its actual class label instead of removing only the majority samples.

3.3 Performance Measures

A confusion matrix, as shown in Table 3, is typically used to evaluate the performance of two-class classification problems (Batista et al., 2004). The columns represent the actual classes while the rows represent the predicted classes. This representation makes it easier to visualize whether instances are being misclassified. Throughout this dissertation, the minority samples represent the developing CCs and the majority samples represent the non-developing CCs.

Table 3

Format of a two-class confusion matrix

		Actual	
		Minority	**Majority**
Predicted	**Minority**	TP	FP
	Majority	FN	TN

The four important parameters found in a two-class confusion matrix are true positive (TP), false positive (FP), false negative (FN), and true negative (TN). In this dissertation, TP represents the number of developing CCs correctly classified, FP represent the number of non-developing CC misclassified as developing CCs, FN represents the number of developing CCs

misclassified as non-developing, and TN represents the number of non-developing CC correctly classified (Bunkhumpornpat, Sinapiromsaran, & Lursinsap, 2012; Chawla, 2005). These four parameters assist in deriving significant performance measures.

When using balanced data, classification problems usually use accuracy as a performance measure (Barua et al., 2014; Batista et al., 2004; Chawla, 2005; Chawla et al., 2002; Si Chen, Guo, & Chen, 2010; García et al., 2007; He & Garcia, 2009). This measure is defined by

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} = 1 - \text{Err}$$

$$Err = \frac{FP + FN}{TP + FP + FN + TN}$$

Accuracy and error rate, Err, are not good performance measures for imbalanced data because it is strongly biased to favor the majority (negative) class. For example, if 98% of a given dataset are majority samples and all samples are classified as being members of the majority class, this would provide an accuracy of 98% (or error rate of 2%). This seems satisfactory but in reality it fails to identify any of the minority samples.

Chawla (2005) suggests that recall, precision, F_1-measure, geometric mean and the area under the receiver operating characteristic (ROC) curve are more suitable for imbalanced data. Recall is also known as sensitivity and the probability of detection (POD). It is used to evaluate the number of minority samples correctly classified. Precision is also known as the positive predictive value (PPV). It measures the number of samples classified correctly as the minority class (Bekkar & Alitouche, 2013; Chawla et al., 2002; He & Garcia, 2009). These two metrics have an inverse relationship. When using these metrics, the goal is to improve the recall without hindering the precision (Chawla, 2005).

$$Recall = Sensitivity = \text{POD} = \frac{TP}{TP + FN}$$

$$Precision = PPV = \frac{TP}{TP + FP}$$

On the other hand, specificity, also known as the true negative rate (TNR), is used to assess the number of majority samples classified correctly and the negative predictive value (NPV) measures the number of samples classified correctly as the majority class. The measures are the opposites of recall and precision.

$$Specificity = TNR = \frac{TN}{TN + FP}$$

$$NPV = \frac{TN}{TN + FN}$$

F_1-Measure (also known as F_1-Score) is a performance measure utilized to evaluate the success of the classification (Bekkar, Djemaa, & Alitouche, 2013; He & Garcia, 2009). It is defined by

$$\text{F}_1\text{-Measure} = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision}$$

A large F_1-Measure value gives a high performance of the minority class. This measure originated from

$$\text{F}_\beta\text{-Measure} = \frac{(1 + \beta^2) \cdot Recall \cdot Precision}{\beta^2 \cdot Recall + Precision}$$

where β changes the significance of precision versus recall. In most cases, recall and precision are equally important therefore β typically equals one (Bekkar et al., 2013; Bunkhumpornpat et al., 2012).

Geometric mean (G-mean) is a performance measure used to assess the balanced performance between the majority and minority classes (Barua et al., 2014; Bekkar et al., 2013; He & Garcia, 2009; Sun, Wong, & Kamel, 2009). This measure is defined by

$$\text{G-Mean} = \sqrt{Recall \cdot Specificity}$$

G-mean is independent of the class distribution and it takes into account the biases of the performance of the minority and majority classes (García et al., 2007). This measure provides a better representation of the performance of an imbalanced problem while it incorporates both the TP rate and the TN rate (Bekkar et al., 2013; He & Garcia, 2009; Sun et al., 2009).

A graphical representation using the TP rate as a function of the FP rate is the ROC curve. This graph is insensitive to class distribution (Barua et al., 2014). The comparison of multiple ROC curves is difficult to assess especially when one curve does not clearly dominate the others. Therefore, it is favorable to obtain a numerical representation of the graph known as the area under the ROC curve (AUC) where $0 \leq AUC \leq 1$ (Barua et al., 2014; Bekkar et al., 2013; Chawla, 2005; Sun et al., 2009). Better classification performance is indicated by larger AUC values.

There are other performance measures that are typically used to access the performance of forecasts and predictions such as Matthew's correlation coefficient (MCC), Heidke skill score (HSS), and threat score (TS). MCC is a performance measure which considers the accuracies and error rates of both classes which is defined by

$$MCC = \frac{TN \cdot TP - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

where $MCC \in [-1,1]$. When MCC is -1, 0, or 1 then the predictions are respectively the worst possible, random, or perfect (Bekkar et al., 2013). The HSS is used to evaluate the performance of a rare event problem. It is an appropriate measure to determine the predictive skill relative to making random guesses (Hennon, Marzban, & Hobgood, 2005; Kerns & Chen, 2013; Wilks, 2005). The HSS is defined by

$$HSS = \frac{2(TN \cdot TP - FP \cdot FN)}{(TN + FN)(FN + TP) + (TN + FP)(FP + TP)}$$

where $HSS \in [-1,1]$. This skill score yields perfect predictions when $HSS = 1$, random predictions when $HSS = 0$, and $HSS < 0$ indicates the predictions have no skill. TS is a statistical measure of the statistical power of the chosen classifier. It is typically useful when analyzing rare events such as developing CCs. The TS, also called the critical success index, measures the fraction of majority events that are correctly predicted. TS is calculated as

$$TS = \frac{TP}{TP + FP + FN}$$

where $TS \in [0,1]$. A perfect forecast occurs when $TS = 1$ and a highly skilled forecast occurs when $TS \geq 0.5$ (Hennon, 2003; Wilks, 2005).

3.4 Summary

This chapter presented a review of techniques for feature selection, classifying imbalanced data, including the contributed oversampling technique SCOT, and commonly used performance measures to evaluate classification problems. These topics are essential to identifying predictive features of developing CCs of a highly imbalanced dataset containing thousands of observations. The succeeding chapter will discuss the overall methodology of identifying predictive features of developing CCs.

CHAPTER 4

Methodology

The purpose of this dissertation is to use feature extraction and oversampling techniques to identify predictive features of CCs that are developing into TCs. The procedure for identifying the predictive features comprises of obtaining the readily accessible satellite data, identification and tracking of each cloud cluster, and distinguishing between developing and non-developing cloud clusters using sampling, feature selection, and classification techniques. This procedure is summarized in Figure 6.

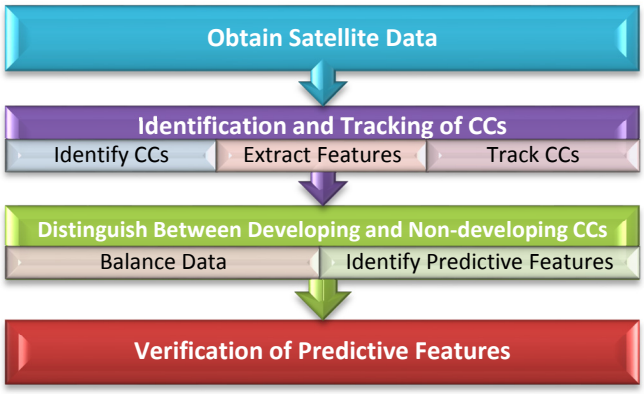


Figure 6. Procedure for identifying the predictive features.

4.1 Software Programs

Few software programs are used for visualization and computations for this research. Those programs are Exelis Visual Information Solutions' Interactive Data Language (IDL) and MathWorks Incorporated's Matlab. IDL is a scientific programming language choice of scientists and engineers, especially in the meteorology or climatology field. This software assists in interpreting data and is used to create visualizations from complex numerical data (Exelis Visual Information Solutions, 2014). In this dissertation, IDL is used to identify individual CCs, extract cloud features, and track CC movements from the obtained satellite data. The CC

features and obtained tracking information are then used as inputs to computations implemented in Matlab.

Matlab is a well-known high-level language that performs computationally exhaustive tasks faster than other programming languages such as C, C++, and Fortran. Matlab is commonly used by engineers and is used in a wide range of applications including control design, computational biology, and signal and image processing (MathWorks Incorporated, 2005). In this research, Matlab is used for computation of feature selection, oversampling, and pattern recognition techniques to help identify predictive features, which can distinguish between developing and non-developing CCs.

4.2 Datasets

The accessibility of information via the internet allowed the acquisition of all required data for this research. Descriptions of the datasets obtained from the National Oceanic and Atmospheric Administration's (NOAA's) National Climatic Data Center (NCDC) are provided.

4.2.1 Hurricane satellite data. NOAA's NCDC provides access to the Hurricane Satellite data (HURSAT–B1, version 05). The HURSAT data comprises of global TC observations from 1978 through 2009. The HURSAT observations have a spatial span of ~10.5^{o} from the center of the observed storm, a temporal resolution of 3 hours, and a gridding resolution of 8 km. The HURSAT data are in a network common data form (netCDF) format and have three available channels: a visible (VIS) channel at 0.65 μm, an IR channel at 11 μm, and an IR water vapor (WV) channel at 6.7 μm (Knapp & Kossin, 2007). Each netCDF file contains a snapshot of a storm from a geostationary weather satellite. The IR channel of the HURSAT data is used to identify and obtain the location of developed TCs. Labelling a CC as developing or

non-developing is dependent on TCs identified by this dataset. Table 4 provides additional specifications regarding the HURSAT-B1 data.

Table 4

Detailed specification of HURSAT and GridSat data (Knapp & Kossin, 2007)

Product	HURSAT-B1	HURSAT-AVHHR	HURSAT-MW	GridSat
Temporal span	1978 – 2009	1978 – 2009	1988 – 2009	1979 - 2009
Spatial span	Storm-centric: 10.5º from center for all global TCs	Storm-centric: 10.5º from center for all global TCs	Storm-centric: 10.5º from center for all global TCs	Global
Temporal resolution	3 hourly	Varying (6 – 12 hourly)	Varying (6 – 12 hourly)	3 hourly
Gridding resolution	8km	4km	8km	8km
Data source	ISCCP B1	AVHRR GAC	DMSP SSM/I	ISCCP B1
Channels available	IRWIN(11μm) IRWVP(6.7μm) (0.65μm)	All AVHRR channels	All SSM/I channels	IRWIN(11μm) IRWVP(6.7μm) (0.65μm)
Calibration	Clim.–IRWIN, ISCCP–IRWVP	Climate calibrated	Operational calibration	Clim.–IRWIN, ISCCP-IRWVP
Yearly size (GB)	< 6.5	40 – 60	4	200
Format	NetCDF	NetCDF	NetCDF	NetCDF
Current version	4.0	Beta	Beta	Beta
Imagery	Movies	BD Imagery	Imagery	Planned

4.2.2 Gridded satellite data. NOAA's NCDC provides access to the Gridded Satellite (GridSat) data. The temporal and gridding resolution of the GridSat data are the same as the

HURSAT data but it includes global observations from 1979 through 2009. For this research, only the IR channel is used because it senses the Earth's surface under clear sky conditions, cloud top temperature of thick clouds, and a combination of cloud and surface temperatures. Both datasets are derived from the International Satellite Cloud Climatology Project (ISCCP) B1 data (Knapp et al., 2011). The GridSat data are used to identify and track all CCs in the atmosphere. Table 4 provides additional specifications regarding the GridSat data.

4.2.3 Reynolds sea surface temperature. NOAA's NCDC provides access to High-Resolution SST blended data with observations from the Advanced Very High Resolution Radiometer (AVHRR) IR satellite. This dataset is derived through optimum interpolation, has a daily temporal resolution and a gridding resolution of 0.25º (Reynolds et al., 2007). This dataset provides the SST corresponding to each CC as a feature in our dataset.

4.3 Identification and Tracking of Cloud Clusters

All CCs that formed above the equator and south of 40ºN in the North Atlantic Ocean are found by examining the 1999-2005 Atlantic hurricane seasons (June 1 – November 30). Throughout this dissertation, all times are reported in a standard Greenwich Time called Coordinated Universal Time (UTC or Z). The UTC times along with its US time zone equivalents are displayed in Table 5.

Table 5

Coordinated Universal Time with its equivalent times in each of the United States time zones

UTC Time	Pacific	Central	Eastern
00	4pm	6pm	7pm
03	7pm	9pm	10pm
06	10pm	12am	1am
09	1am	3am	4am

Table 5

Cont.

12	4am	6am	7am
15	7am	9am	10am
18	10am	12pm	1pm
21	1pm	3pm	4pm

4.3.1 Identification of cloud clusters. As discussed in Chapter 2, there are multiple definitions of a CC. Therefore, based on previous studies a formal definition of a CC is established to identify CCs objectively. Overall, a CC should have the ability to developing into a TC. Therefore, the CC must have sufficient BTs, sufficient size, must persist for a prolonged period of time, and must exist in an area where genesis is possible which is typically not in high latitudes. For this study, the following criterion is used to objectively identify CCs:

I. A cluster must be located to the south of 40^{o}N

II. A cluster must last for at least 24 hours

III. A cluster must have a BT less than or equal to 250 K (-23.15^{o}C)

IV. A cluster must have an area of at least 5,000 km^2

Prior studies use a colder BT threshold and a larger area threshold (Futyan & Del Genio, 2007; Hennon et al., 2011; Machado et al., 1998). Instead, more conservative thresholds are used to account for CCs that may convert to a warm core system, to account for CCs that may change rapidly in size, and to avoid the need to apply fixes for missing data.

Once a CC is identified using the GridSat data, each CC is given a serial number for reference. The serial number follows the format YYYYMMDDHHLsLaLaLaLpLoLoLoLo where YYYY, MM, DD, and HH represent the year, month, day, and hour respectively. Ls

denotes the direction (north or south) of the latitude coordinate LaLaLa and Lp denotes the direction (east or west) of the longitude coordinate LoLoLoLo. The actual latitude coordinates are in the format LaLa.La with a range of 0.0 to 90.0º and the actual longitude coordinates are in the format LoLoLo.Lo with a range of 0.0 to 180.0º. Each coordinate contains one decimal place and the decimals are removed to fit in the serial number format. For example, a CC located at (25.95ºW, 10.96ºN) at 15Z August 4, 2000 is assigned 2000080415N109W0259 as its serial number.

In the CC identification stage, a few variables are assigned to provide additional information regarding the CC. These variables are: *Time*, *StormName*, and *LandFlag*. *Time* specifies the time of the CC observation in the format YYYYMMDDHH, which denotes the year, month, day, and hour of the CC respectively. *StormName* is defined as "NA" if the CC is not a developed TC. Otherwise, it is assigned the corresponding storm name from the HURSAT dataset. The *LandFlag* variable indicates whether the CC is over land or ocean. The formation of a TC typically occurs over an ocean basin and, in this case, $LandFlag = 0$. In cases where a CC develops into a TC, we continue to obtain information regarding the developed TC regardless of if it makes landfall ($LandFlag = 1$). Developed TCs are the only CCs in our dataset where $LandFlag = 1$.

4.3.2 Cloud cluster feature extraction. The features extracted from each CC are separated into four different categories: location, shape, statistical, and image. There are 9 location features that provide information on the location of each CC. The location features can become valid predictive features if there is an adequate separation between the spatial distribution of developing and non-developing CCs. Figure 7 and Figure 8 illustrate the spatial distribution of the geometric center $(\mathrm{glon}, \mathrm{glat})$ of the first observation of all non-developing and

developing CCs. There is not a visual distinction between the locations of the developing and non-developing CCs. There is simply a higher occurrence of non-developing CCs than developing CCs.

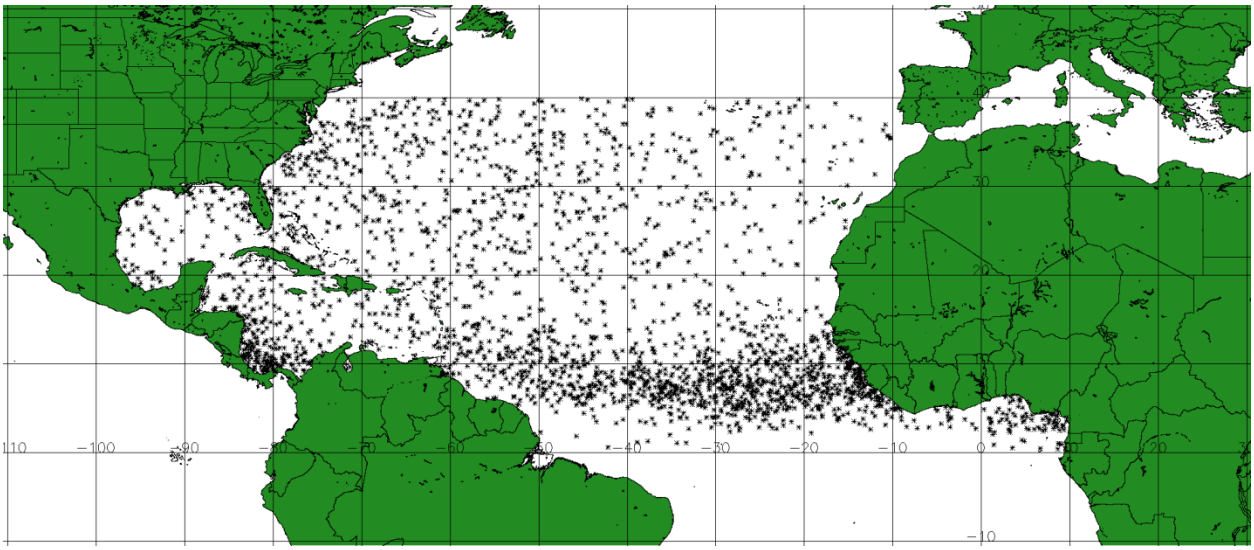


Figure 7. Spatial distribution of the first observation of all non-developing cloud clusters for the 1999-2005 North Atlantic hurricane seasons.

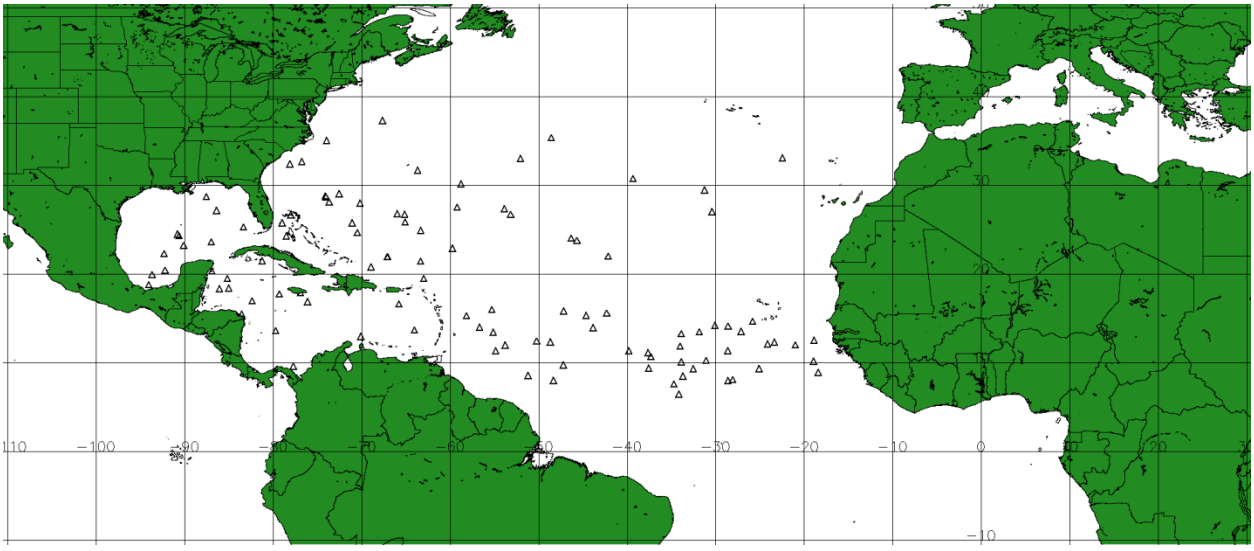


Figure 8. Spatial distribution of the first observation of all developing cloud clusters for the 1999-2005 North Atlantic hurricane seasons.

In addition to location features, there are 13 shape features, which provide information about the shape of the CC, 50 statistical features that use the BT to obtain information about the CC, and 8 image features that contain information regarding the relationship of the pixels in an image with a spatial span of approximately 10.5º from the center of the observed CC. The 50 statistical features consist of 36 features that are based on the mean and standard deviation of the BTs, and the minimum BT for 12 rings that are in 50 km increments from 50 km to 600 km from the CC center. There are also five statistical features which indicate the percentage of CC pixels that are less than or equal to 195 K (-78.15ºC), 205 K (-68.15ºC), 215 K (-58.15ºC), 225 K (-48.15ºC), and 235 K (-38.15ºC). The feature variables listed in Table 6 are extracted from each CC. Additional information regarding equations and descriptions of each feature is located in Appendix A. These features are evaluated to determine which predictive features contribute to the development of a TC.

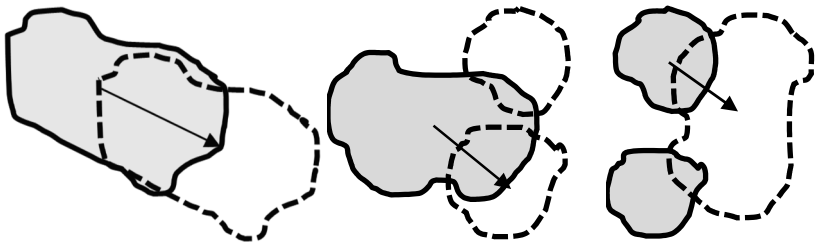
Table 6

List of features extracted from each identified cloud cluster

Location (9)	**Shape (13)**		**Statistical (50)**		**Image (8)**
$ALAT17$	$A\ (km)$	λ_1	BT_{avg}	$RingBT_{minXX}$	EBC
d_{TC}	$A\ (pixels)$	λ_2	SST_{avg}	$RingBT_{stdXX}$	H_b
$glon$	Com	G	BT_{kurt}	BT_{std}	Con
$glat$	Ecc	Ro	BT_{skew}	BT_{min}	Cor
$mlon$	E_{va}		$BT_{5\%}$		E
$mlat$	R_{Est}		$BT_{10\%}$		ECF
SC	R_{Max}		Fc		Hom
$wlon$	R_{Min}		BTP_{XXX}		NMI
$wlat$	P		$RingBT_{avgXX}$		

4.3.3 Tracking cloud clusters. Once each CC is identified and its corresponding features are extracted, they are then tracked to trace their evolution. The approach used to track individual CCs incorporate the area overlap method. This technique assumes that a CC at time t corresponds to a CC at time $t + 1$ if there are common pixels in consecutive satellite images and the size and the BT criterion are met. This method is a relatively simple technique that is commonly used since it tracks CCs based on consecutive observations. When tracking CCs, it is important to account for the splitting and merging occurrences; therefore, it is possible for an overlap to exist for multiple CCs. Five possible conditions can occur when using this tracking method.

1) **Generation:** Occurs when there is not a CC present at time t but there is a CC present at time $t + 1$. This represents the beginning of a new CC.
2) **Dissipation:** Occurs when there is a CC present at time t but there is not a CC present at time $t + 1$. This represents the dissipation of a CC.
3) **Continuance:** Occurs when there is an overlap of only one pair of CCs as shown in Figure 9a. In this figure, the gray CCs represent time t and the white dotted CCs represent time $t + 1$.
4) **Split:** Occurs when a CC at time t overlaps multiple CCs at time $t + 1$ as shown in Figure 9b. The CC interaction with the larger overlap is typically chosen to continue the CC evolution and all other CCs represent a generation of a new CC.
5) **Merge:** This situation is the opposite of a split. A merge occurs when multiple CCs at time t overlap with a single CC at time $t + 1$. An example of this case is shown in Figure 9c. The CC interaction with the larger overlap is chosen to continue the CC evolution and all other CCs represent a dissipating CC.



(a) (b) (c)

Figure 9. Schematic representation of (a) continuing, (b) splitting, and (c) merging cloud clusters. The gray figures represent a cloud cluster at time t and the white dotted figures represent a cloud cluster at time $t + 1$. The arrows represent the actual evolution of the cloud cluster.

To determine which CC interaction represents the best CC track in the splitting and merging cases, the overlap of sequential CCs is calculated by the maximum scaled overlap SO_{max} which is defined as

$$SO_{max} = \frac{CC_{\mathrm{t}} \cap CC_{\mathrm{t+1}}}{\max(A_{\mathrm{t}}, A_{\mathrm{t+1}})}$$

where A_{t} and $A_{\mathrm{t+1}}$ denote the area of the CCs at time t and $\mathrm{t} + 1$, respectively. If multiple CC interactions have the same SO_{max} value, then the interaction with the highest minimum scaled overlap SO_{min} is selected. Minimum scaled overlap is defined as

$$SO_{min} = \frac{CC_{\mathrm{t}} \cap CC_{\mathrm{t+1}}}{\min(A_{\mathrm{t}}, A_{\mathrm{t+1}})}$$

Once the identification and tracking of all CCs is complete, the characteristics of each CC is contained in a multivariate time series. Each time series is labeled as a developing or non-developing CC dependent on the developed TCs identified using the HURSAT data.

Identification and tracking of all CCs and their extracted features are the most important contribution of this study because there is no ground truth dataset. However, there are numerous

CCs in the atmosphere and the techniques must be accurate and completed in an objective manner so individuals other than forecasters can use them. Therefore, we validated the proposed methods by comparing our tracks of developed TCs to those recorded in the HURSAT dataset.

Figure 10 shows an example of the HURSAT centers and the calculated centers (geometric, weighted, and minimum BT) for Hurricane Cindy (1999). As shown, the calculated centers vary from the HURSAT centers because the calculated centers are always inside the CC. Therefore, these centers are automatically calculated based on the shape and/or BT of the CC. On the other hand, the HURSAT centers are subjective and their centers are not always inside a CC. The differences in the centers demonstrate the benefits of our research, which is based solely on observations and are not subjective.

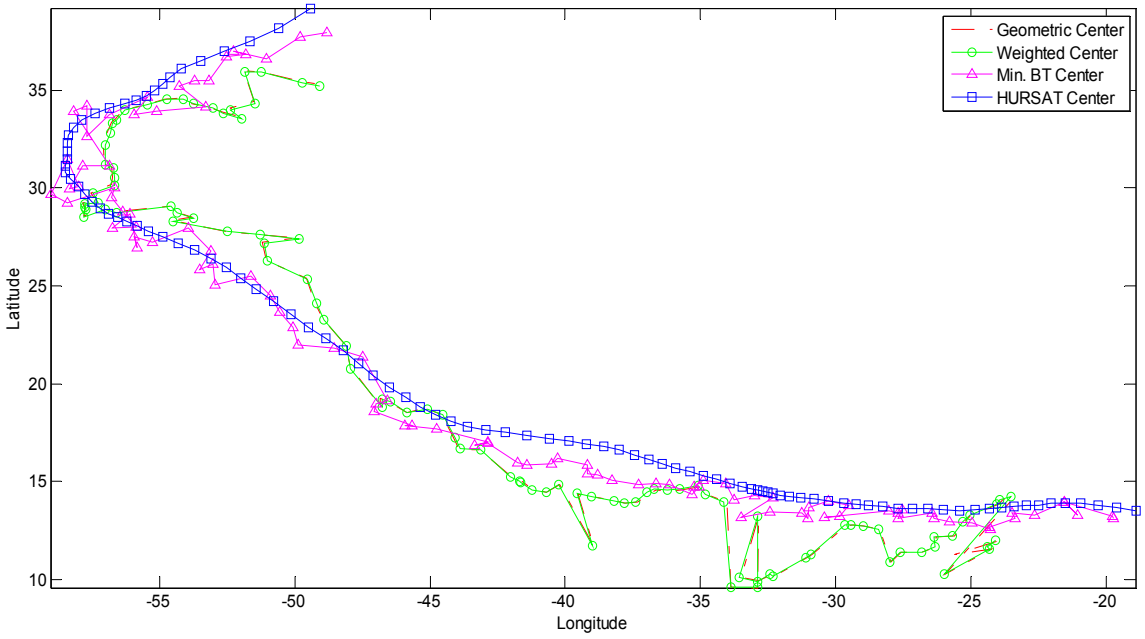


Figure 10. Plot of HURSAT centers and calculated centers for Hurricane Cindy (1999).

4.3.4 Characteristics of cloud cluster feature dataset. The number of North Atlantic TCs in the HURSAT dataset are not always equivalent to the number of TCs in our CC feature dataset because we eliminate any CC that persist for less than 24 hours or are pole ward of 40ºN.

These CCs are eliminated because they do not abide by our CC criterion found in Section 4.3.1. Statistics of the CCs that met the criterion for this dissertation during the seven North Atlantic Hurricane seasons are presented in Table 7. The 2005 season is the most active in developing CCs while the 1999 season was the most active in non-developing CCs. The characteristics from Table 7 are not equivalent to prior studies because our BT and area thresholds in the identification process are more conservative. Hence, having conservative thresholds identify more CCs and it eliminates the need to apply fixes to our data as done in Hennon et al. (2011) due to their colder BT and larger area thresholds.

Table 7

Summary of cloud clusters for the 1999-2005 North Atlantic hurricane seasons that meet the proposed cloud cluster criterion

	1999	**2000**	**2001**	**2002**	**2003**	**2004**	**2005**	**Overall**
Total # of CCs	521	479	513	480	499	495	525	3512
Longest duration (hours)*	186	138	114	156	120	159	132	186
Mean duration (hours)*	40.06	38.93	39.53	39.34	39.49	39.10	39.02	39.36
Median duration (hours)*	36	33	33	33	33	33	33	33
# of Developing CCs	15	17	19	15	17	14	30	127
% of Developing CCs	2.88	3.55	3.70	3.13	3.41	2.83	5.71	3.62
# of Non-developing CCs	506	462	494	465	482	481	495	3385
% of Non-developing CCs	97.12	96.45	96.30	96.88	96.59	97.17	94.29	96.38

* Non-developing CCs only

4.4 Distinguishing between Developing and Non-developing Cloud Clusters

Identifying predictive features of developing CCs using solely gridded satellite data is a difficult task. When addressing this problem, we must convert CC time series to individual observations, standardize the data, balance the data to make the class distribution approximately equal, identify predictive features, and classify the data. The methods used for these steps are described in this section.

4.4.1 Convert cloud cluster time series. There are multiple ways of representing our CC feature dataset. We convert our CC time series data in a manner that can be used for real time classification in the future. This representation includes all observations of each CC time series dependent on the forecast being analyzed. There are nine forecasts which contain all non-developing observations and observations of developing CCs that occur at 0, 6, 12, 18, 24, 30, 36, 42, and 48 hours prior to the development of a TC. In this representation, a developing CC is categorized as non-developing if it does not develop into a TC within 48 hours.

4.4.2 Standardization of the dataset. After changing the representation of the data, we standardize the data by converting the values to z-scores using the following equation:

$$\widetilde{CC}_j^i = \frac{\left(CC_j^i - \overline{CC}^i\right)}{s^i}$$

where CC_j^i denotes sample j of feature i, $\overline{CC}^i$ represents the sample mean of feature i, and s^i indicates the sample standard deviation of feature i. The data is standardized to avoid confusion during the classification of features that have different magnitudes and units. Therefore, the selected classifier will treat each variable with equal consideration and the standardization can help stabilize the training of the classifier. Before further analysis, observations that contain any missing values were excluded.

4.4.3 Balance cloud cluster data. In most real world applications, the observed data are highly imbalanced which causes a problem since standard classifiers are biased to the larger class. In this research, the observations of non-developing CCs outnumber those of developing CCs. Therefore, to address this issue we eliminate outliers and apply ENN for undersampling, and we apply SCOT for oversampling.

The amount of non-developing CCs greatly outnumbers developing CCs by thousands, which is due to our conservative thresholds. Therefore, to reduce the number of samples, we eliminate mild outliers from the non-developing CCs using the first quartile (Q_1), third quartile (Q_3) and the interquartile range (IQR), which is equivalent to the difference between Q_1 and Q_3 (Lewis, 2012). A CC is a mild outlier if

$$CC_j^i < Q_1 - (1.5 \times IQR)$$

or

$$CC_j^i > Q_3 + (1.5 \times IQR)$$

This method of identifying mild outliers focuses on the positions of the first and third quartile (Lewis, 2012). Once the mild outliers are eliminated, the ENN is used for undersampling as described in Chapter 3. ENN is used because it is expected to remove non-developing CCs that may overlap with the developing CCs causing misclassification.

Once the number of non-developing CCs is reduced, SCOT is used because of its satisfactory performances, which were described in Chapter 3 and Appendix B. Specifically, SCOT assures that the synthetic samples do not replicate any of the minority samples or other synthetic samples. Here, SCOT is used to balance the CC feature data so we can use standard classifiers to determine the best predictive features to identify developing CCs. We analyze nine forecasts, which contain observations that occur at 0, 6, 12, 18, 24, 30, 36, 42, and 48 hours prior

to the development of the TCs. Table 8 provides a comparison of the number of samples for each forecast before and after balancing the dataset. Note that the totals in Table 8 are different from that of Table 7 because it is based on the individual observations instead of the complete time series data. Table 8 shows that the developing CCs on average over all forecasts were approximately 0.96% of the imbalanced dataset but once ENN and SCOT are applied, its population increased to approximately 50.57% of the balanced dataset. Balancing the data verifies that the number of samples in each class are *approximately* equal which reduces the bias of the non-developing CCs (majority class) when using a standard classifier.

Table 8

Comparison of the number of cloud clusters for each forecast before and after balancing the data so the number of samples in each class are approximately equal

Forecast	Characteristic	Before Balancing	After Balancing
0	# of CCs	44997	32743
	# of Developing CCs	1108	16398
	% of Developing CCs	2.46	50.08
	# of Non-Developing CCs	43889	16345
	% of Non-Developing CCs	97.54	49.92
6	# of CCs	44746	33501
	# of Developing CCs	857	16942
	% of Developing CCs	1.92	50.57
	# of Non-Developing CCs	43889	16559
	% of Non-Developing CCs	98.08	49.43
12	# of CCs	44523	33954
	# of Developing CCs	634	17126
	% of Developing CCs	1.42	50.44
	# of Non-Developing CCs	43889	16828
	% of Non-Developing CCs	98.58	49.56

Table 8

Cont.

18	# of CCs	44356	34259
	# of Developing CCs	467	17190
	% of Developing CCs	1.05	50.18
	# of Non-Developing CCs	43889	17069
	% of Non-Developing CCs	98.95	49.82
24	# of CCs	44229	34648
	# of Developing CCs	340	17377
	% of Developing CCs	0.77	50.15
	# of Non-Developing CCs	43889	17271
	% of Non-Developing CCs	99.23	49.85
30	# of CCs	44119	35044
	# of Developing CCs	230	17589
	% of Developing CCs	0.52	50.19
	# of Non-Developing CCs	43889	17455
	% of Non-Developing CCs	99.48	49.81
36	# of CCs	44020	35390
	# of Developing CCs	131	17735
	% of Developing CCs	0.30	50.11
	# of Non-Developing CCs	43889	17655
	% of Non-Developing CCs	99.70	49.89
42	# of CCs	43951	35681
	# of Developing CCs	62	17864
	% of Developing CCs	0.14	50.07
	# of Non-Developing CCs	43889	17817
	% of Non-Developing CCs	99.86	49.93
48	# of CCs	43904	35874
	# of Developing CCs	15	17939
	% of Developing CCs	0.03	50.01
	# of Non-Developing CCs	43889	17935
	% of Non-Developing CCs	99.97	49.99

4.4.4 Identification of predictive features. To select an appropriate subset of features (or predictors) for developing CCs, the SFS algorithm was used as described in Chapter 3. In this case, SFS selects the predictors using a stopping criterion for the feature selection process of 0.0001, and using a selected classification performance measure. The classifier chosen to assess the SFS process is logistic regression. Binary logistic regression is a statistical method that analyzes a dataset containing features and binary class labels equal to 1 for developing CCs and 0 for non-developing CCs. We used three different classification performance measures to assist in the SFS method. These measures are the average of sensitivity and specificity (Acc), HSS, and G-Mean whose definitions are found in Chapter 3.

Table 9

Comparison of sequential forward selection using different performance measures and their classification results for the CART simulation

Performance Measure	Forecast	# of Features	Features	F_1-Measure	G-Mean	HSS
Acc	0	9	ALAT17 SST_{avg} H_b d_{TC} Ecc mlat NMI $RingBT_{std550}$ BT_{std}	99.97499	99.97498	0.9995
	6			99.98863	99.98863	0.99977
	12			99.96702	99.96702	0.99934
	18			99.80666	99.80655	0.99613
	24			99.96816	99.96816	0.99936
	30			99.7988	99.79859	0.99597
	36			99.94313	99.94313	0.99886
	42			99.94314	99.94313	0.99886
	48			99.97726	99.97725	0.99955
HSS	0	4	SST_{avg} d_{TC} BTP_{195K} $RingBT_{std200}$	99.95793	99.95792	0.99916
	6			99.98636	99.98636	0.99973
	12			99.92838	99.92836	0.99857
	18			99.79973	99.79974	0.99599
	24			99.94771	99.94769	0.99895
	30			99.63654	99.63579	0.99272
	36			99.89881	99.89876	0.99798
	42			99.94882	99.94882	0.99898
	48			99.95906	99.95906	0.99918

Table 9

Cont.

GMean	0	6	ALAT17 SST_{avg} d_{TC} $BT_{5\%}$ mlat NMI	99.97839	99.97839	0.99957
	6			99.98408	99.98408	0.99968
	12			99.95566	99.95565	0.99911
	18			99.76236	99.76216	0.99524
	24			99.97044	99.97044	0.99941
	30			99.76583	99.7656	0.99531
	36			99.91926	99.91924	0.99838
	42			99.95223	99.95223	0.99904
	48			99.96589	99.96588	0.99932

The classification results for a simple classification and regression trees (CART) simulation using ten-fold cross validation for all forecasts are displayed in Table 9. This simulation classifies all CC observations using the selected predictive features, which were identified using SFS and various performance measures. As shown in the table, each performance measure selects different features as predictors but all results are satisfactory. To determine which SFS method is significant, we use the Wilcoxon Signed Rank test to compare the G-Mean values (ideal performance measure) of SFS using G-Mean to the results of SFS using HSS and Acc. This test is used because it is a powerful non-parametric test (Sheng Chen, He, & Garcia, 2010; Demsar, 2006). For each forecast, we calculate the difference d_i between the G-Mean values of the compared methods. The absolute values of the differences are ranked from least to greatest where the smallest difference obtains a ranking of 1. An average rank is given if a tie occurs in more than one difference. The sign (+ or -) of the difference is applied to each ranking and summed based on their signs as follows (Demsar, 2006):

$$R^+ = \sum_{d_i>0} \text{rank}(|d_i|) + \frac{1}{2}\sum_{d_i=0} \text{rank}(|d_i|)$$

$$R^- = \sum_{d_i<0} \text{rank}(|d_i|) + \frac{1}{2}\sum_{d_i=0} \text{rank}(|d_i|)$$

Table 10

Wilcoxon Signed Rank test results, which compare the geometric means of the sequential forward selection methods using geometric mean, heidke skill score, and average of sensitivity and specificity as performance measures

Forecast	SFS using G-Mean vs. HSS			SFS using G-Mean vs. Acc		
	G-Mean	HSS	Rank	G-Mean	Acc	Rank
0	**99.97839**	99.95792	4.0	**99.97839**	99.97498	2.0
6	99.98408	99.98636	-1.0	99.98408	**99.98863**	-3.0
12	99.95565	99.92836	7.0	99.95565	**99.96702**	-5.5
18	99.76216	99.79974	-8.0	99.76216	**99.80655**	-9.0
24	**99.97044**	99.94769	6.0	**99.97044**	99.96816	1.0
30	99.7656	99.63579	9.0	99.7656	**99.79859**	-8.0
36	99.91924	99.89876	5.0	99.91924	**99.94313**	-7.0
42	**99.95223**	99.94882	2.0	**99.95223**	99.94313	4.0
48	**99.96588**	99.95906	3.0	**99.96588**	99.97725	5.5
	T = min{36, 9} = 9			T = min{12.5, 32.5} = 12.5		

To determine the significance of using the SFS method, a T value is computed as $T = min(R^+, R^-)$. The null hypothesis of this test is that the considered methods perform equally. In order to reject this hypothesis, the T value must be less than or equal to its relative critical value found in a Wilcoxon Signed Rank critical value table such as the one found in Bissonnette

(2011). From the Wilcoxon Signed Rank critical value table we find that for nine forecasts and a significance level of 0.05 ($\alpha = 0.05$), the value of T should be less than or equal to 5 if the difference between the methods are significant (Demsar, 2006). Table 10 displays the results of this test with the best results highlighted in bold. The table concludes that we cannot reject the null hypothesis when comparing SFS using G-Mean versus Acc or HSS, which indicates that the three methods perform equally. Hence, we identify the predictors as the union of the features selected by the methods. Based on our feature selection results, our dataset is reduced to only include the selected predictors. The dimensions of our dataset are reduced drastically once we use the predictors instead of using all eighty features. The reduction in the number of dimensions increases the speed and reduces the computation time of our classification process.

4.5 Summary

This chapter presented the methodology for this research. Our methods used IDL to objectively identify and track CCs, and to extract eighty features for each CC from the obtained satellite data. On the other hand, Matlab was used for feature selection, oversampling, and pattern recognition techniques to help identify the best set of predictive features obtained from applying SFS. The succeeding chapter will discuss the identified predictive features and the classification results of using the predictive features.

CHAPTER 5

Verification of Predictive Features

Identifying developing TCs is a complicated task; therefore, it is beneficial to identify predictive features that can objectively identify developing CCs from global gridded satellite data. Identifying predictive features may improve the performance of identifying developing CCs by eliminating redundant features which may reduce classification performance and have high computational costs. The preceding chapter provided the methodology for identifying the predictive features. Based on the results of the Wilcoxon Signed Rank test, we identify the predictors as the union of the features selected after applying SFS using G-Mean, HSS, and Acc. The twelve features identified as predictors consist of three location features, one shape feature, six statistical features, and two image features with the age of the CC being an additional parameter. The following are the selected predictors: latitude of maximum genesis productivity, distance to nearest TC, average latitude of the minimum BT, eccentricity, average SST, BT in which 5% of the CC pixels are colder, percentage of CC pixels less than 195 K (-78.15^{o}C), standard deviation of BT in rings with a radius of 200 and 550 km from the geometric center, standard deviation of BT, binary entropy, and normalized moment of inertia. These predictors indicate that some information regarding all feature types (location, shape, statistical, and image) is required to successfully identify developing CCs from solely gridded satellite data.

5.1 Verification of Predictive Features using Standard Classifiers

The goal of presenting simulation results is to evaluate the performance of identifying developing CCs using the identified predictive features. To verify that our simulations are not classifier dependent, one specific classifier is not used. Instead, we apply CART, a simple neural network, and a support vector machine (SVM) using ten-fold cross validation to classify the

simplified 1999-2005 CC feature dataset containing only the identified predictors and the age parameter. These classifiers are used because the CART algorithm is one of the simplest algorithms that does not require additional parameters, a neural network is more complex, and both the CART algorithm and the neural network can provide probabilistic forecasts (Shao & Lunetta, 2012). On the other hand, the SVM does not provide probabilistic forecasts but this classifier is used primarily for its performance in real world problems, its generalization capability, and its fast and effective learning (Gavrishchaka & Ganguli, 2001; Shao & Lunetta, 2012). Therefore, satisfactory results for all simulations demonstrate the ability of our techniques.

5.1.1 Optimal design of probabilistic classifiers. When using a neural network, it is imperative that the network is designed to yield optimal results (Doukim, Dargham, & Chekima, 2010; Hennon, 2003; Sheela & Deepa, 2013). Therefore, an optimal number of neurons in the hidden layer for each forecast hour is determined. Many studies have evaluated many techniques of determining an optimal number of hidden neurons and most techniques consider three rule-of-thumb methods

1. $H = O + (0.75I)$,
2. $H \leq 2I$, and
3. $O \leq H \leq I$

where O, H, and I represent the number of neurons in the output, hidden, and input layers, respectively (Karsoliya, 2012; Shahamiri & Binti Salim, 2014; Sheela & Deepa, 2013). Therefore, we consider possible values of hidden neurons between one and twenty-six for each forecast hour since our input layer contains thirteen neurons, and our output layer contains one neuron. The CC feature dataset is analyzed for ten trials using 10-fold cross validation for all

possible of hidden neurons values. Thus, the network was trained 2600 times for each forecast hour.

The optimal number of hidden neurons for each forecast hour is determined in the following manner. For each trial, the dataset is classified using a neural network with the scaled conjugate gradient backpropagation as the training function, the hyperbolic tangent sigmoid transfer function as the input activation function, the log-sigmoid transfer function as the output activation function, and the mean squared error as the performance function. The number of hidden neurons with minimal error is selected for each of the ten trials. Once all trials are concluded, ten values of possible hidden layer sizes are suggested. The hidden layer size that occurs more frequently for the considered forecast hour is chosen as the optimal number of hidden neurons. Figure 11 is a frequency histogram of the optimal number of hidden neurons for all forecast hours. The optimal number of hidden neurons for each forecast hour are as follows: 25 neurons for the 0 hour forecast, 23 neurons for the 6 hour forecast, 23 neurons for the 12 hour forecast, 24 neurons for the 18 hour forecast, 25 neurons for the 24 hour forecast, 25 neurons for the 30 hour forecast, 22 neurons for the 36 hour forecast, 22 neurons for the 42 hour forecast, and 23 neurons for the 48 hour forecast.

The performance measures used to evaluate the simulations are based on the confusion matrix. This matrix is generated based on discrete forecasts of 1 (developing) or 0 (non-developing). Therefore, an optimal decision threshold (D_{th}) is ideal where probability of forecasts above (below) the threshold are classified as developing (non-developing) CCs and the skill of the forecasts are maximized (Hennon, 2003). To identify D_{th} for each forecast hour, the entire CC feature dataset was classified using the corresponding classifier and 10-fold cross validation. The probabilistic forecasts were evaluated using the confusion matrix for all possible

decision thresholds between 0 and 1 in 0.001 increments. The decision threshold obtaining the highest HSS value is selected as D_{th}. In instances where there are equivalent HSS values, the lower decision threshold is selected. The optimal decision thresholds for each of the forecast hours are indicated in Table *11* and are used hereafter.

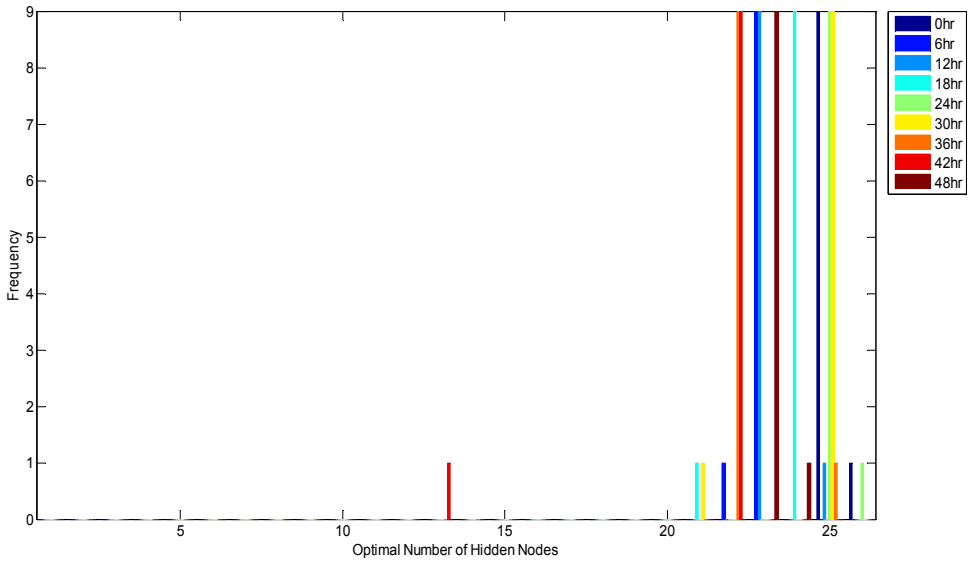


Figure 11. Histogram of the optimal number of hidden neurons for the 90 trials.

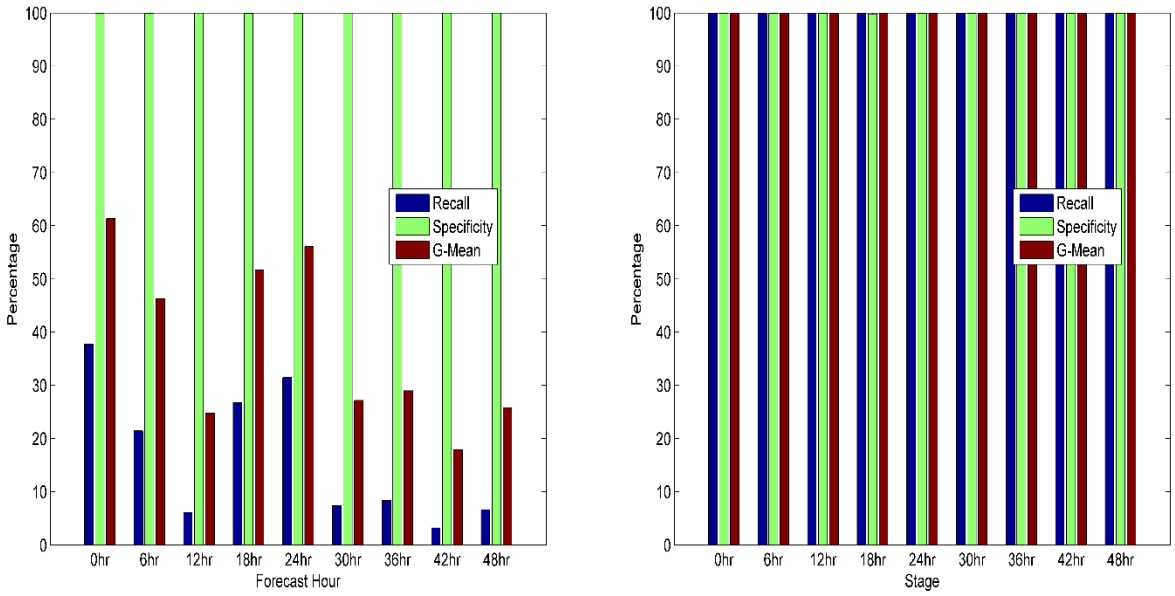
Table 11

Optimal decision thresholds for each forecast hour

	0	**6**	**12**	**18**	**24**	**30**	**36**	**42**	**48**
CART	0.700	0.800	0.750	0.667	0.500	0.400	0.667	0.667	0.048
Neural Network	0.527	0.549	0.590	0.591	0.644	0.698	0.778	0.858	0.886

5.1.2 Classification and regression trees. CART is one of the original techniques for classification problems which uses a tree structure where the leaves represent class labels (developing or non-developing CCs) and the branches signify combinations of the predictive

features that result in those labels (Narsky & Porter, 2013; Shao & Lunetta, 2012). The CART algorithm is implemented using the Statistics Toolbox in Matlab to distinguish between developing and non-developing CCs. The classifier is implemented to use pruning, have at least one observation per tree leaf, uses the Gini's diversity index as a splitting criterion, merge leaves that originate from the same parent node, and the class probabilities are based on the class distribution. Figure 12 illustrates the classification performance using the identified predictive features and the performance measures for each forecast in the simulation are found in Table 12.



(a) (b)

Figure 12. Comparison of the geometric means and the performance of developing (recall) and non-developing (specificity) cloud clusters using (a) the imbalanced dataset and (b) the balanced dataset for each forecast hour for the CART simulation.

When the dataset is imbalanced, the CART classifier has poor performance in identifying developing CCs. Its best performance occurs at the 0 hour forecast where only 37.73% and 99.85% of the developing and non-developing CCs are identified correctly. This is expected

because the classifier is bias to the non-developing CCs since they represent majority of the data. The precision performance measure provides a representation of the amount of developing CCs that are correctly classified as developing. Therefore, the imbalanced dataset has low recall and higher precision values which means there are few non-developing CCs being misclassified as developing. This indicates that the CART algorithm is fairly confident when it classifies a CC as developing.

Table 12

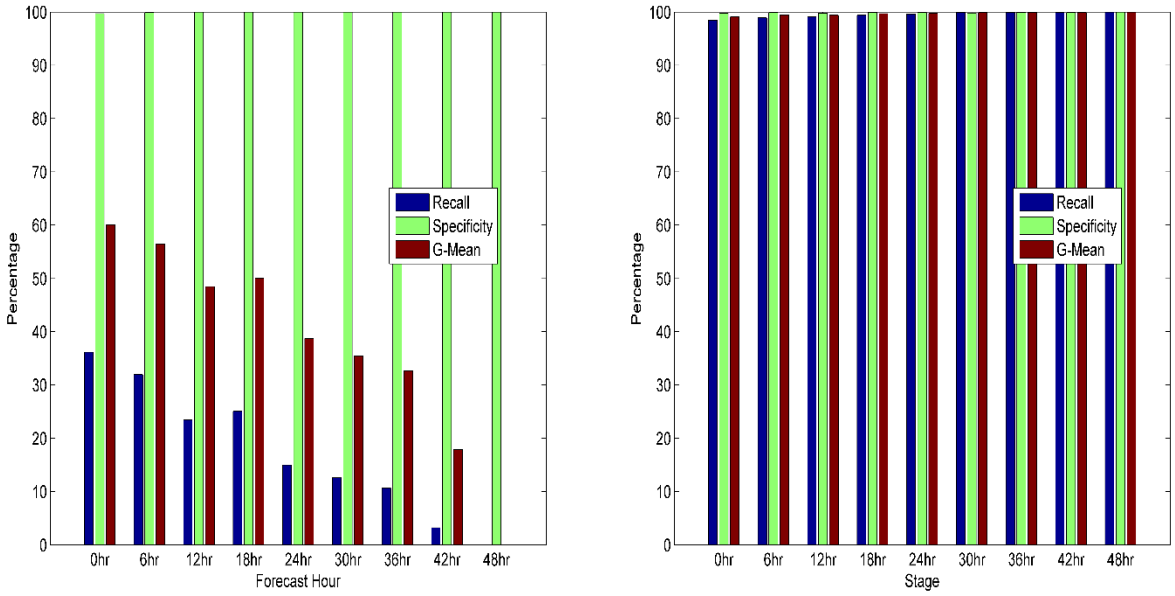
Performance measures for each forecast for the CART simulation

Dataset	Forecast	Recall	Specificity	Precision	F_1-Measure	G-Mean	HSS	TS
Imbalanced	0	37.72563	99.85317	94.57014	53.93548	61.37608	0.52205	0.36926
	6	21.47025	99.94565	95.33679	35.04762	46.3234	0.33851	0.21247
	12	6.15142	100	100	11.5899	24.80206	0.11216	0.06151
	18	26.7666	99.94141	92.59259	41.52824	51.72129	0.40821	0.26205
	24	31.47059	99.91894	88.42975	46.42082	56.07591	0.45872	0.30226
	30	7.3913	100	100	13.76518	27.18695	0.13611	0.07391
	36	8.39695	99.99434	91.66667	15.38462	28.97666	0.1528	0.08333
	42	3.22581	100	100	6.25	17.96053	0.0623	0.03226
	48	6.66667	99.99442	50	11.76471	25.81917	0.11747	0.0625
Balanced	0	99.85822	99.83216	99.84188	99.85005	99.84519	0.99691	0.99701
	6	99.85724	99.85047	99.85724	99.85724	99.85386	0.99708	0.99715
	12	99.8508	99.87992	99.88392	99.86736	99.86536	0.9973	0.99735
	18	99.88357	99.81241	99.81716	99.85036	99.84799	0.99697	0.99701
	24	99.91663	99.83588	99.83894	99.87777	99.87625	0.99753	0.99756
	30	99.9109	99.86474	99.86641	99.88865	99.88782	0.99776	0.99778
	36	99.92198	99.85415	99.8552	99.88858	99.88806	0.99776	0.99777
	42	99.9108	99.83788	99.83845	99.87461	99.87434	0.99749	0.9975
	48	99.94423	99.83824	99.83845	99.89131	99.89122	0.99782	0.99783

It is difficult to identify the developing CCs since there are few samples. The distribution of the developing and non-developing CCs is approximately equal after applying SCOT because SCOT generates synthetic samples for the developing CCs. Generating synthetic samples increases the ability to identify developing CCs. This is demonstrated by the performance of the balanced dataset where at least 99.85% and 99.81% of developing and non-developing CC are correctly identified. Therefore, all evaluated forecasts obtain a geometric mean of at least 99.85% which indicates the confidence in the identified predictive features. All of the performance measures for the balance dataset demonstrate satisfactory results for identifying developing CCs specifically with high recall and high precision values.

5.1.3 Neural network. Neural networks are typically used because it does not make assumptions regarding the distribution of the data (Shao & Lunetta, 2012). Therefore, this simulation uses a simple neural network implemented using the Neural Network Toolbox in Matlab. The implemented neural network has one hidden layer, uses the scaled conjugate gradient backpropagation as the training function, the hyperbolic tangent sigmoid transfer function as the input activation function, the log-sigmoid transfer function as the output activation function, and the mean squared error as the performance function (Beale, Hagan, & Demuth, 2013). Figure 13 illustrates the performance of using the identified predictive features to classify developing and non-developing CCs in the neural network simulation. As illustrated in this figure, the performance of identifying non-developing CCs is high while the performance of identifying developing CCs decreases for longer forecasts when the dataset is imbalanced. Once the class distribution is approximately equal, all evaluated forecasts obtain a geometric mean above 99.09 % which indicates the identified predictive features can satisfactorily identify developing CCs. In addition to high geometric mean values, high recall and precision values

indicate the ability to identify developing CCs without misclassifying non-developing CCs as developing. The calculated performance measures for each forecast are found in Table 13. This table demonstrates SCOT's skill of increasing the ability to identify developing CCs using the identified predictive features.



(a) (b)

Figure 13. Comparison of the geometric means and the performance of developing (recall) and non-developing (specificity) cloud clusters using (a) the imbalanced dataset and (b) the balanced dataset for each forecast for the neural network simulation.

Table 13

Performance measures for each forecast for the neural network simulation

Dataset	Forecast	Recall	Specificity	Precision	F_1-Measure	G-Mean	HSS	TS
Imbalanced	0	36.19134	99.66351	87.9386	51.27877	60.05793	0.49406	0.3448
	6	31.972	99.87922	93.19728	47.61077	56.50963	0.4626	0.31243
	12	23.50158	99.97029	96.75325	37.81726	48.47122	0.36922	0.23318
	18	25.05353	99.99414	99.15254	40	50.05204	0.39348	0.25
	24	15	99.99421	98.07692	26.02041	38.72871	0.2564	0.14956
	30	12.6087	100	100	22.39382	35.50873	0.22167	0.12609
	36	10.68702	100	100	19.31034	32.69101	0.19195	0.10687

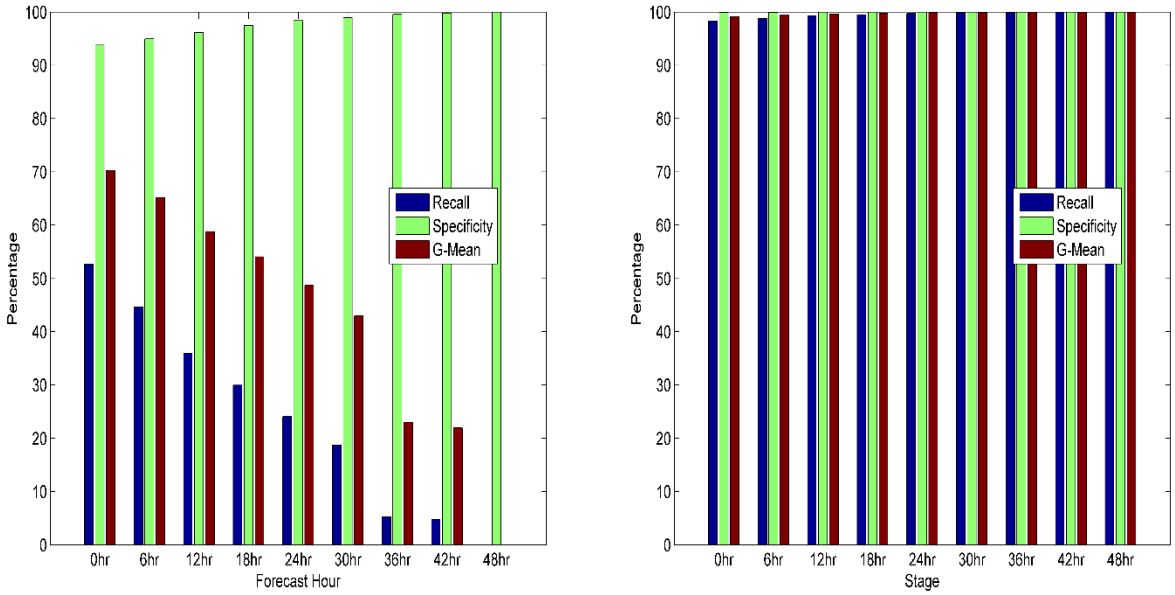
Table 13

Cont.

	42	3.22581	100	100	6.25	17.96053	0.0623	0.03226
	48	0	100	NaN	NaN	0	0	0
Balanced	0	98.42949	99.7627	99.77337	99.09687	99.09386	0.98152	0.9821
	6	98.90188	99.89648	99.90017	99.39852	99.39794	0.98775	0.98804
	12	99.07167	99.76555	99.77184	99.42052	99.41801	0.98825	0.98848
	18	99.40677	99.93179	99.93312	99.66925	99.66894	0.99332	0.99341
	24	99.63317	99.86418	99.8663	99.7496	99.74861	0.99495	0.995
	30	99.84965	99.82528	99.82741	99.83853	99.83747	0.99675	0.99678
	36	99.91083	99.94951	99.94982	99.93033	99.93017	0.9986	0.99861
	42	99.92195	99.86024	99.86072	99.89133	99.89109	0.99782	0.99783
	48	99.98327	99.98327	99.98327	99.98327	99.98327	0.99967	0.99967

5.1.4 Support vector machine. The SVMs are typically used because they are effective in high dimensional spaces and they perform well on sparse and noisy data. This classifier separates the data with a maximally distant hyperplane in the feature space which can satisfactorily separate the developing and non-developing CCs (Furey et al., 2000). It is implemented using the Statistics Toolbox in Matlab. The SVM is designed to use the Gaussian radial basis function as the kernel function with a default scaling factor (sigma) of one, the maximum number of iterations to converge is set to 100,000, and the sequential minimal optimization algorithm is used to find the separating hyperplane. Figure 14 illustrates the classification performance using the identified predictive features and the performance measures for each forecast in the simulation are found in Table 14. When the dataset is imbalanced, at least 93.77% of the non-developing CCs are identified correctly while the identification of developing CCs does not perform as well. The imbalanced dataset has low recall and low precision values which means many non-developing CCs are being misclassified as developing. This indicates that the SVM algorithm has limited confidence when it classifies a CC as developing. On the other hand, at least 98.35 % and 99.93 % of developing and non-developing

CC are correctly identified once the dataset is balanced. Therefore, all evaluated forecasts obtain a geometric mean of at least 99.09% which indicates the ability of identifying developing CCs using only the identified predictive features and the age parameter.



(a) (b)

Figure 14. Comparison of the geometric means and the performance of developing (recall) and non-developing (specificity) cloud clusters using (a) the imbalanced dataset and (b) the balanced dataset for each forecast for the support vector machine simulation.

Table 14

Performance measures for each forecast for the support vector machine simulation

Dataset	Forecast	Recall	Specificity	Precision	F_1-Measure	G-Mean	HSS	TS
Imbalanced	0	52.70758	93.76568	36.43169	43.08373	70.30051	0.38464	0.27457
	6	44.69078	94.95139	31.4192	36.89788	65.14178	0.33028	0.22623
	12	35.96215	96.17304	26.14679	30.27888	58.80977	0.27219	0.1784
	18	29.97859	97.47495	24.51839	26.97495	54.05702	0.24771	0.1559
	24	24.11765	98.37879	22.65193	23.36182	48.71001	0.21805	0.13226
	30	18.69565	98.95159	19.02655	18.85965	43.01121	0.178	0.10412
	36	5.34351	99.56386	8.33333	6.51163	23.06557	0.0597	0.03365
	42	4.83871	99.82601	8.82353	6.25	21.97792	0.06019	0.03226

Table 14

Cont.

	48	0	99.98327	0	NaN	0	-.00028	0
Balanced	0	98.35315	99.93055	99.93351	99.13703	99.13871	0.98236	0.98289
	6	98.79756	99.94824	99.95001	99.37044	99.37124	0.98719	0.98749
	12	99.25402	99.98856	99.98887	99.62009	99.62061	0.9923	0.99243
	18	99.50657	99.99432	99.99443	99.7499	99.75014	0.99495	0.99501
	24	99.69987	99.98868	99.98885	99.84415	99.84417	0.99686	0.99689
	30	99.88306	99.98873	99.98885	99.93593	99.93588	0.99871	0.99872
	36	99.9387	99.99439	99.99442	99.96655	99.96654	0.99933	0.99933
	42	99.96655	99.99441	99.99442	99.98049	99.98048	0.99961	0.99961
	48	100	99.99442	99.99442	99.99721	99.99721	0.99994	0.99994

5.2 Summary

This chapter presents the simulation results to verify the performance of the identified predictive features. For all forecasts and simulations, the results indicate a F1-Measure of at least 99.09%, a geometric mean above 99.09%, a HSS of at least 0.98, and a TS above 0.98. These values are a huge improvement when compared to the results without using SCOT to make the class distribution of developing and non-developing CCs approximately equal. Figure 15 and Figure 16 demonstrate the improvement of the G-Mean values and the HSSs, respectively, for all forecast hours using SCOT when compared to not using SCOT. When SCOT is not applied and the dataset is imbalanced, longer forecast hours decrease in predictive skill. This indicates that SCOT increases the size of the minority class (developing CCs) without hindering the ability to distinguish non-developing CCs from those that will develop. Overall, the results show that the selected predictors from our CC feature dataset and the application of SCOT can satisfactorily separate developing and non-developing CCs.

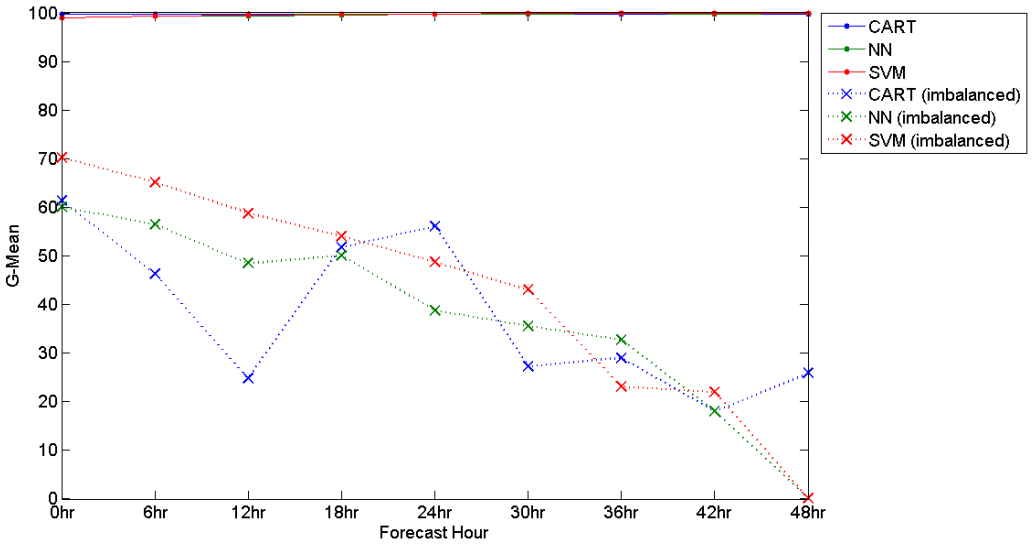


Figure 15. Comparison of the geometric mean of the imbalanced and the balanced datasets for all forecast hours and classifiers.

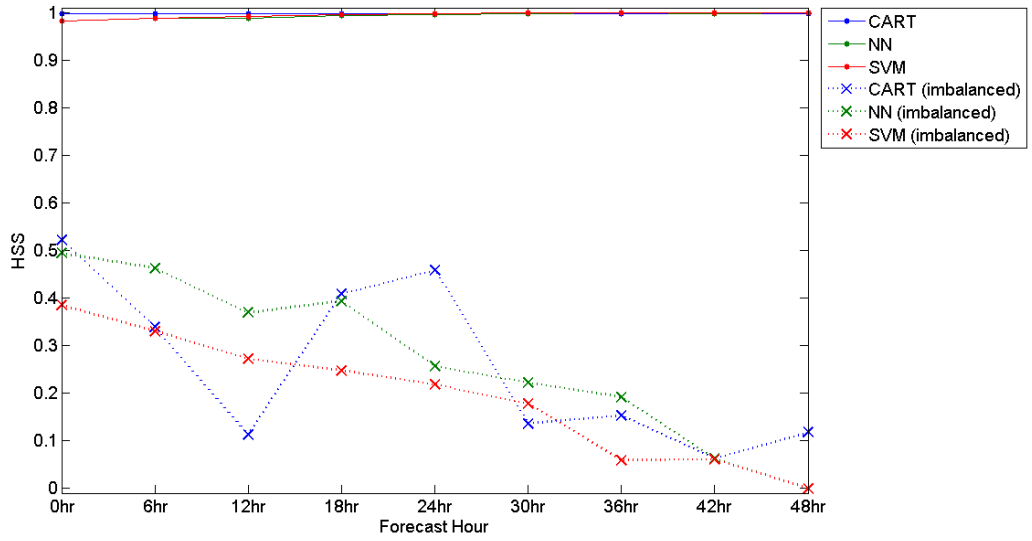


Figure 16. Comparison of the Heidke skill score of the imbalanced and the balanced datasets for all forecast hours and classifiers.

The results of three different classifiers are provided in this chapter. Each classifier has satisfactory performance for the balanced dataset but the main difference in the classifiers are

visible by its performance on the imbalanced dataset. The CART algorithm is the simplest of the compared classifiers that does not require additional parameters and is computationally inexpensive in comparison to the compared classifiers. This classifiers has low recall values and high specificity values for the imbalanced dataset but the precision values are at least 88.43% with the exception of the 48 hour forecast. The neural network classifier is more complex than CART and it requires the specification of additional parameters. The performance is dependent on the specified additional parameters therefore they must be chosen in an objective manner. This classifiers has low recall values and high specificity values for the imbalanced dataset but the precision values are at least 87.94% with the exception of the 48 hour forecast. The SVM classifier is known for its generalization capability and its fast and effective learning capabilities. This is partially demonstrated by the results but this classifier was the most computationally expensive classifier of the three. This classifiers has low recall and precision values and high specificity values for the imbalanced dataset. The low precision and recall values demonstrates less confidence in the decisions of identifying a CC as developing. Based on the overall results of each classifier, the preferred classifier is the neural network classifier because it can be optimally designed, it performs well on complex data, and its results are more consistent as demonstrated by the steadiness in the G-Mean and HSS values in Figure 15 and Figure 16, respectively. The succeeding chapter evaluates case studies to further verify the performance of our techniques.

CHAPTER 6

Case Studies

The performance of multiple classifiers is evaluated on the 1999-2005 North Atlantic hurricane season in the preceding chapter. The CART, the neural network, and the SVM classifiers all have satisfactory performance of identifying developing CCs. The results demonstrate a drastic increase in performance when comparing the imbalanced data results to those of the balanced dataset especially since its ability does not decrease for longer forecasting hours. In this chapter we examine seven case studies to further verify the performance of our techniques. The first six case studies are within the 1999-2005 dataset and the last case study is from the 2006 North Atlantic hurricane season to further verify that our techniques performs well with other datasets. These studies were randomly selected, with an exception of the historic Hurricane Katrina (2005), and evaluated using the same classifiers from the previous chapter. The National Hurricane Center website provided summaries for the developing case studies. The data associated with each case study were removed from the dataset and used as the test samples while the remaining data were used as the training samples. Therefore, for each case study there were twenty seven different datasets (9 forecasts per classifier).

The following case studies are shown using an index value called TCG Index (TCGI). This scale produces an index ranging from -1 (least favorable) to 1 (most favorable) and is defined as follows:

$$TCGI = \frac{P - D_{th}}{1 - D_{th}} \text{ if } P \geq D_{th}$$

$$TCGI = \frac{P - D_{th}}{D_{th}} \text{ if } P < D_{th}$$

where P denotes the probability from the classifier and D_{th} represents the optimal decision threshold for the respective classifier and forecast as indicated in Table *11*. For the SVM classifier, P is either 1 (developing) or 0 (non-developing) and $D_{th} = 0.5$. Figure 17 through Figure 19 display histograms of TCGI values for developing and non-developing CCs for the CART, neural network, and SVM simulations, respectively. These figures show that the TCGI values for developing and non-developing CCs are clearly separable for all simulations. This suggests our techniques can satisfactorily identify developing CCs. The succeeding sections focus on the TCGI values of the selected case study.

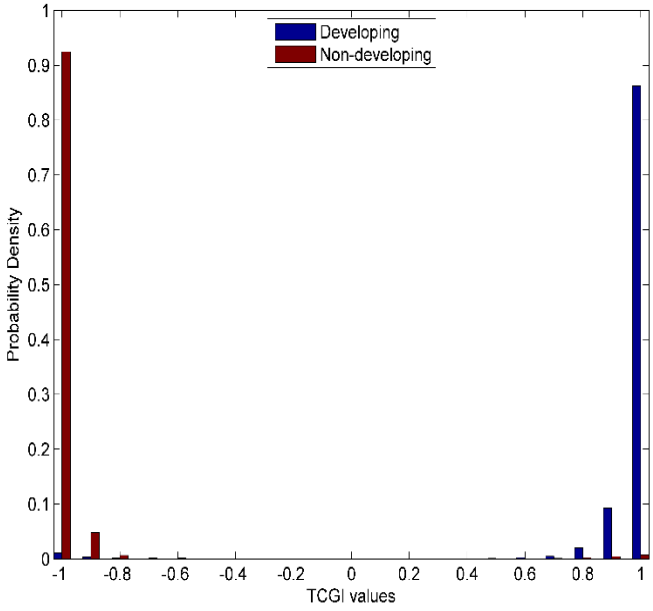


Figure 17. Histogram of the TCGI values for developing and non-developing CCs for the CART simulation.

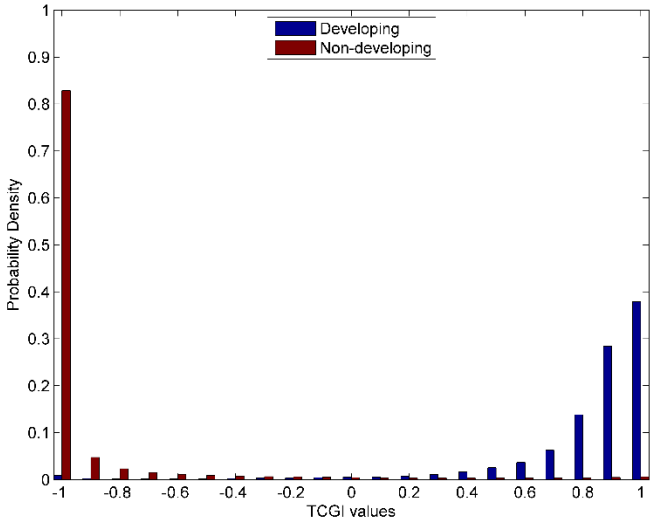


Figure 18. Histogram of the TCGI values for developing and non-developing CCs for the neural network simulation.

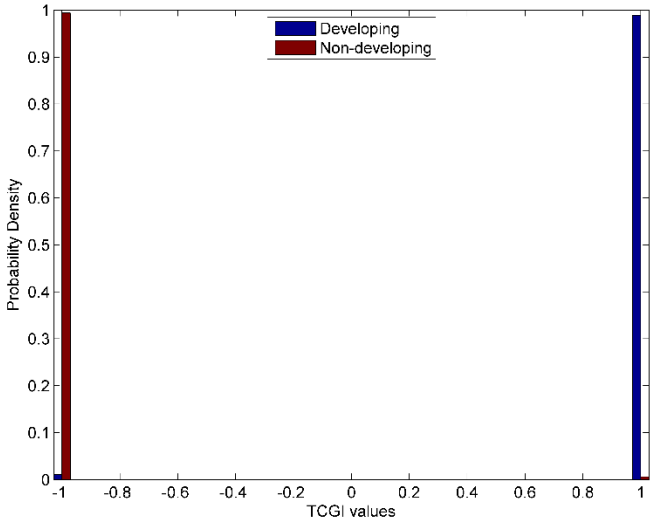


Figure 19. Histogram of the TCGI values for developing and non-developing CCs for the support vector machine simulation.

6.1 Hurricane Katrina (2005)

In the CC dataset, the CC that eventually became Hurricane Katrina originated at 12Z August 21 with a geometric center at (20.79ºN, 68.95ºW). This system originated from the

interaction of an upper tropospheric trough over the Bahamas, remnants of Tropical Depression Ten, and a tropical wave departing the west coast of Africa (Knabb, Rhome, & Brown, 2005). This interaction produced a large region of convection and thunderstorms which slowly progressed northwestward during August 22. This slow movement coincides with the data in the CC dataset. By 18Z August 23, the system was declared a tropical depression at (23.1ºN, 75.1ºW) which was 36 hours after the Tropical Weather Outlook (TWO) began conveying the possibility of the CC developing (Knabb et al., 2005). Figure 20 displays the CC at genesis from our CC dataset and from the HURSAT dataset after applying our BT threshold of 250K. Both figures are images that are 301 by 301 pixels where the center of the storm is the center of the image. The CC from our dataset uses the geometric center as the center while the HURSAT center is specified by an expert. The difference in the defined centers is visible by the shift of the CC in the image.

(a) (b)

Figure 20. Hurricane Katrina at 18Z August 23 from (a) our CC dataset and (b) the HURSAT data after applying our brightness temperature threshold.

A series of forecasts were analyzed for each developing and non-developing CC of Katrina using the same classifiers presented in the preceding chapter. Figure 21 displays the TCGI values for each forecast hour for Hurricane Katrina (2005) for the CART simulation. In this figure, the first forecast to identify the storm as developing is the 36 hour forecast for the CC observed on 3Z August 22. This occurred during the time a slow propagating large region of convection and thunderstorms was produced which was 39 hours prior to its genesis. The CART simulation is one of the simplest classifiers "learned" by splitting the dataset into subsets based on the values of each predictor. Therefore, to examine its overall performance over all forecast hours, the average TCGI values are obtained as displayed in Figure 22. The increase in TCGI values in this simulation indicates that as the CC evolves, genesis is more favorable. For example, the TCGI value from 12Z August 21 to 9Z August 22 are relatively low (< -0.32), with a neutral value at 12Z August 22, and there are oscillations in the TCGI values with a downward trend until an abrupt increase to 0.31 at 3Z August 23.

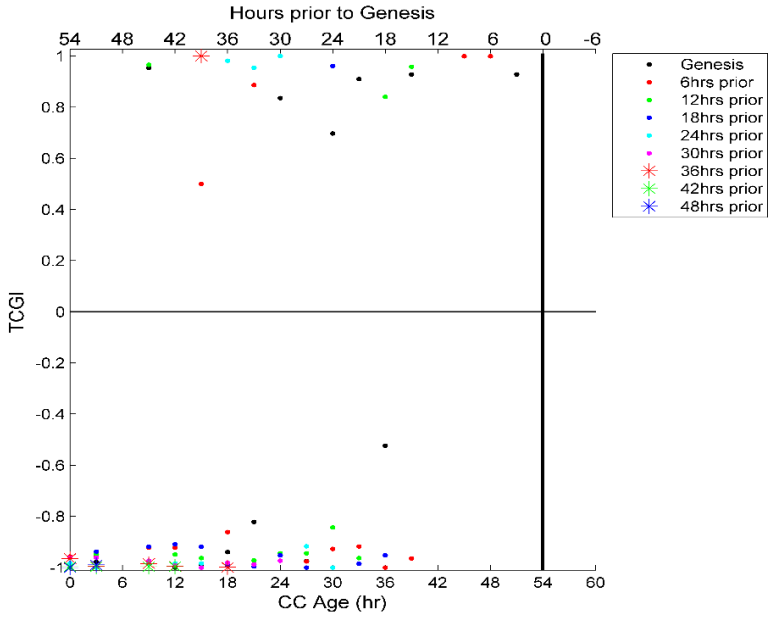


Figure 21. TCGI values for each forecast hour for Hurricane Katrina (2005) for the CART simulation.

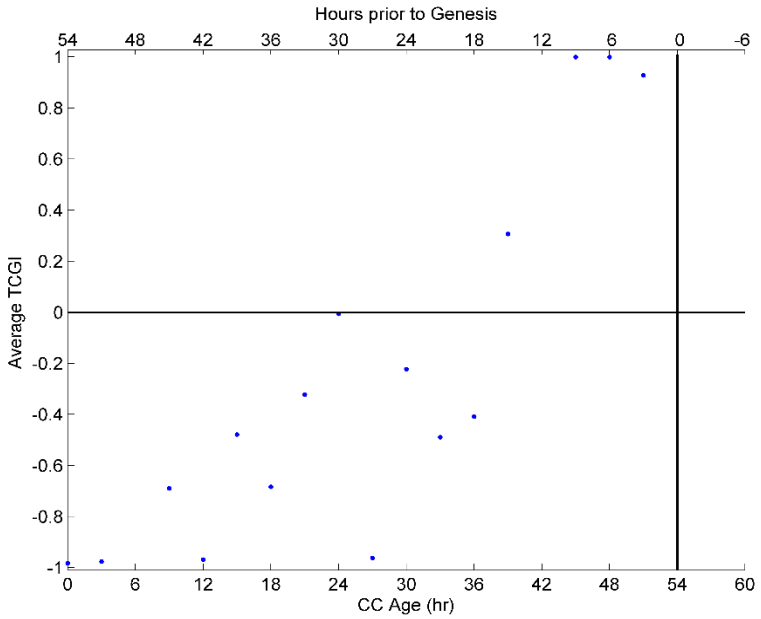


Figure 22. Average TCGI values for Hurricane Katrina (2005) for the CART simulation.

Figure 23 displays the TCGI values for each forecast hour for Hurricane Katrina (2005) for the neural network simulation. In this figure, the first forecast to identify the storm as developing is the 30 hour forecast for the CC observed on 6Z August 22. This was also during the time a slow propagating large region of convection and thunderstorms was produced. The difference between this simulation and the CART simulation is that there are more CC observations identified as favorable for TC development. This is demonstrated by the number of observations above zero and even those CCs identified as non-favorable have higher values than the CART simulation. This change is contributed to the neural network classifier having the ability to recognize more complex patterns in the data. The average TCGI values are obtained as displayed in Figure 24. In this figure, the average TCGI values increase throughout time and is considered favorable for development beginning at 18Z August 22 with the exception of 3Z August 23 (-0.09) which is slightly unfavorable for development.

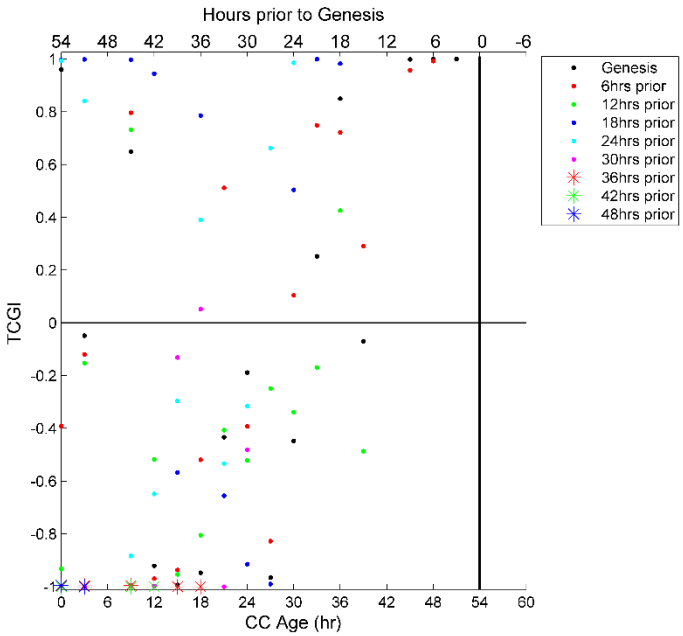


Figure 23. TCGI values for each forecast hour for Hurricane Katrina (2005) for the neural network simulation.

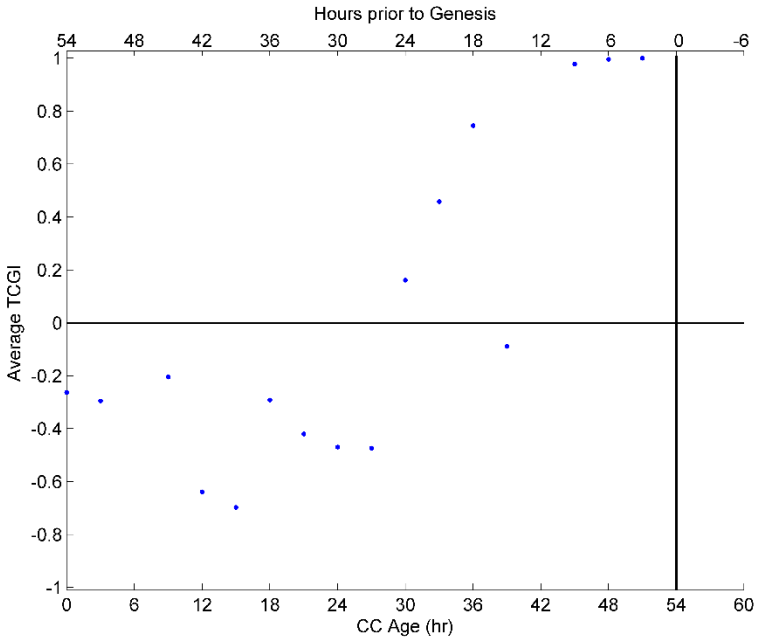


Figure 24. Average TCGI values for Hurricane Katrina (2005) for the neural network simulation.

The TCGI values for each forecast hour for Hurricane Katrina (2005) for the SVM simulation is shown in Figure 25. Since the SVM simulation is non-probabilistic, the TCGI values are either -1 for non-developing or 1 for developing CCs. In this figure, the first forecast to identify the storm as developing is the 36 hour forecast for the CC observed on 3Z August 22 which is the same as the CART simulation. To examine its overall performance over all forecast hours, the average TCGI values are obtained as displayed in Figure 26. Based on these values, the pre-Katrina CC has favorable conditions for development as early as 3Z August 22 with the exception of 9Z August 22, 21Z August 22, 3Z August 23, and 9Z August 23.

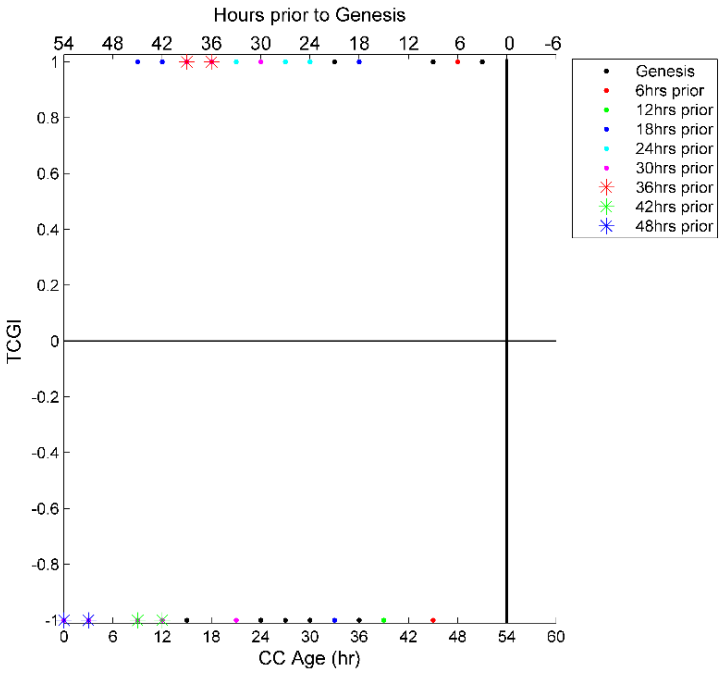


Figure 25. TCGI values for each forecast hour for Hurricane Katrina (2005) for the support vector machine simulation.

Overall, all classifiers performed well during the phases of development. The simulation results indicate that the CART simulation suffers more than the neural network simulation which suffers more than the SVM simulation as supported by the oscillations in the TCGI values

throughout the development phases of the Katrina CCs. When comparing the average TCGI values over all classifiers, Hurricane Katrina can be identified as a developing CC 39 hours prior to development which is at 3Z August 22. The difference in simulations could be attributed to complex interactions present between the predictors which are difficult to address using the "divide and conquer" method implemented in CARTs. On the other hand, neural networks and SVMs can address such interactions better even though SVM is more computationally expensive.

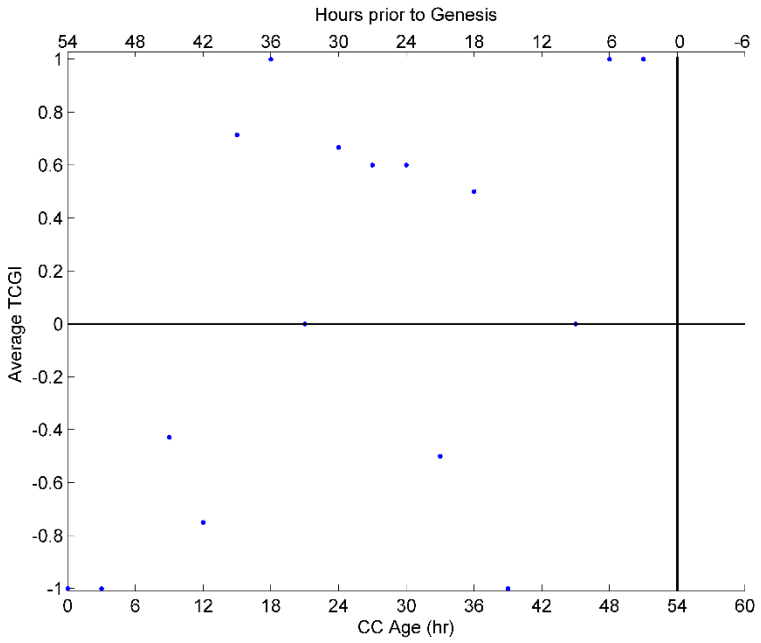


Figure 26. Average TCGI values for Hurricane Katrina (2005) for the support vector machine simulation.

6.2 Hurricane Olga (2001)

The CC that eventually became Hurricane Olga originated at 15Z November 20 with a geometric center at (35.07ºN, 73.96ºW). This system originated during the latter part of the 2001 North Atlantic hurricane season from a cold front and disturbed weather on November 22 between the Leeward Islands and Bermuda (Avila, 2001). The HURSAT data indicates the

genesis of Hurricane Olga occurred at 6Z November 23 while Avila (2001) indicates 0Z November 24. Note that our CC feature dataset uses the HURSAT date as the genesis date. Figure 27 displays the CC at genesis from our CC dataset and from the HURSAT dataset after applying our BT threshold of 250K. At this time, Olga has low circulation, a comma-shaped cloud band, and extends northward which eventually formed circulation as thunderstorm activity increased (Avila, 2001).

(a) (b)

Figure 27. Hurricane Olga at 6Z November 23 from (a) our CC dataset and (b) the HURSAT data after applying our brightness temperature threshold.

The TCGI values for each forecast hour for Hurricane Olga (2001) for the CART, neural network, and SVM simulations are displayed in Figure 28 through Figure 30, respectively. All simulations have the ability of identifying the storm as developing in the 42 hour forecast for the CC observed on 6Z November 21. This occurred before the cold front reached the Bermuda area and the CC obtained tropical characteristics. This indicate that our techniques satisfactorily identified Hurricane Olga. This is also further verified by Figure 31 through Figure 33, which

displays the average TCGI values for all forecast hours for the CART, neural network, and SVM simulations, respectively.

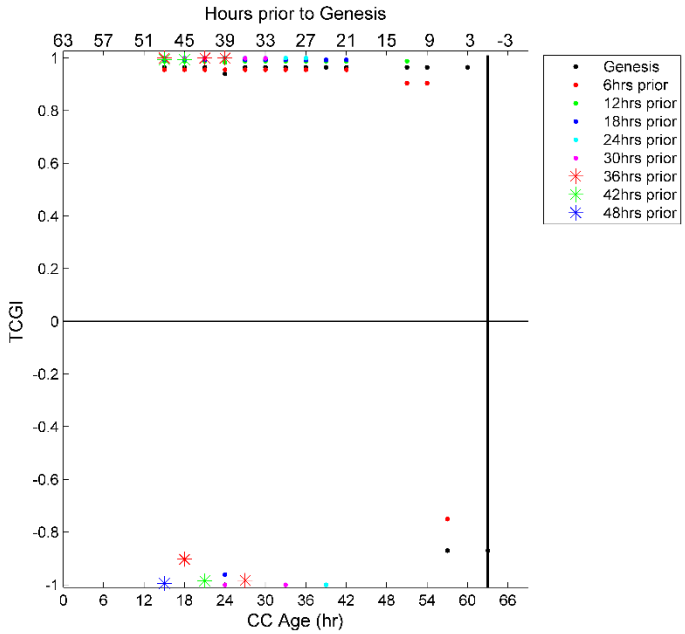


Figure 28. TCGI values for each forecast hour for Hurricane Olga (2001) for the CART simulation.

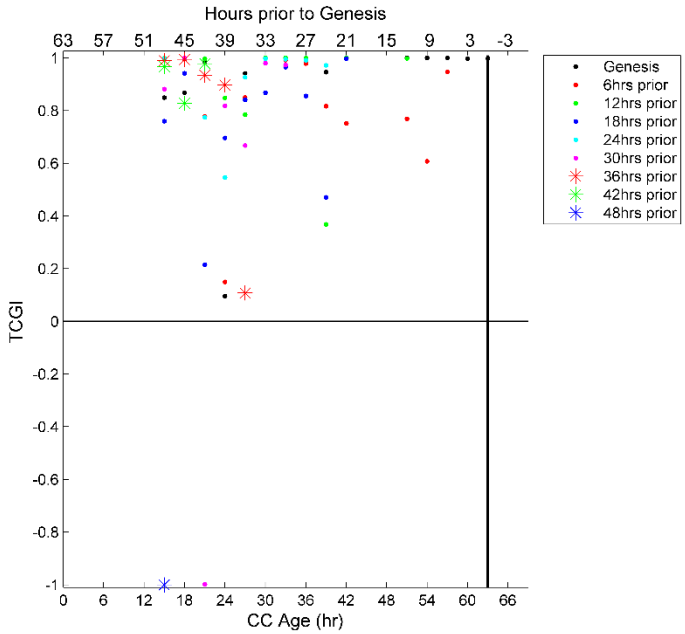


Figure 29. TCGI values for each forecast hour for Hurricane Olga (2001) for the neural network simulation.

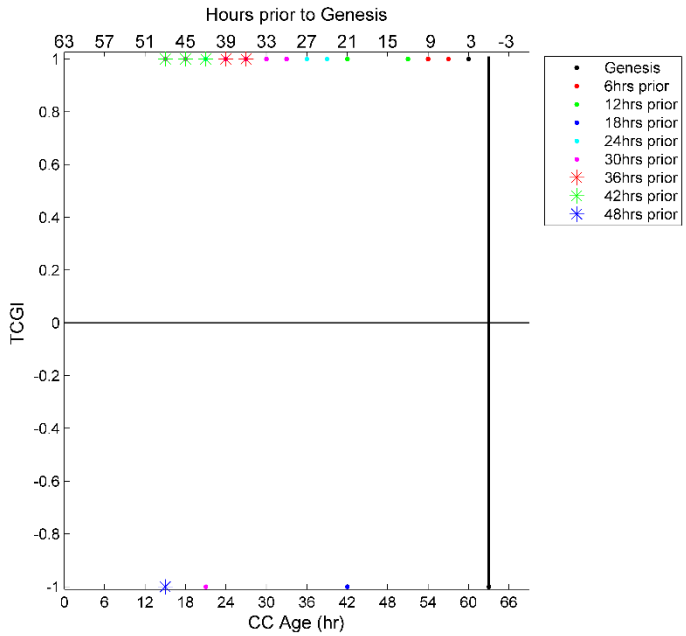


Figure 30. TCGI values for each forecast hour for Hurricane Olga (2001) for the support vector machine simulation.

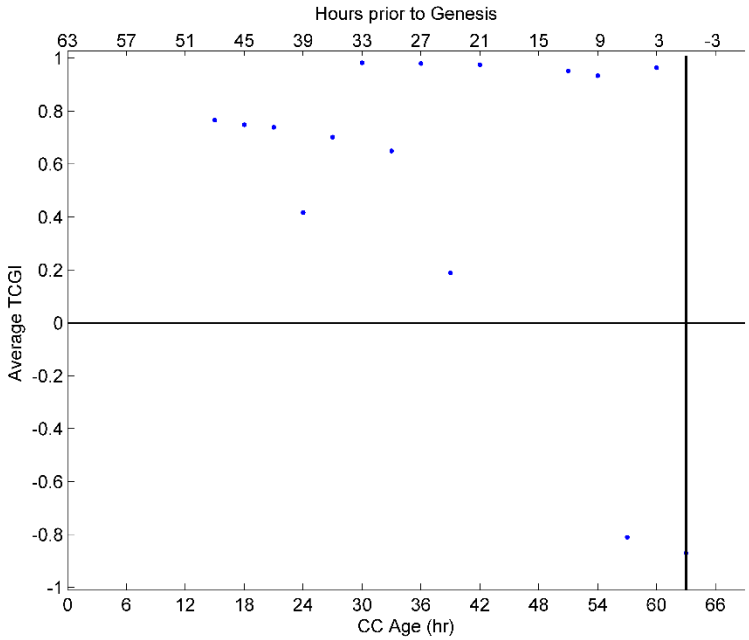


Figure 31. Average TCGI values for Hurricane Olga (2001) for the CART simulation.

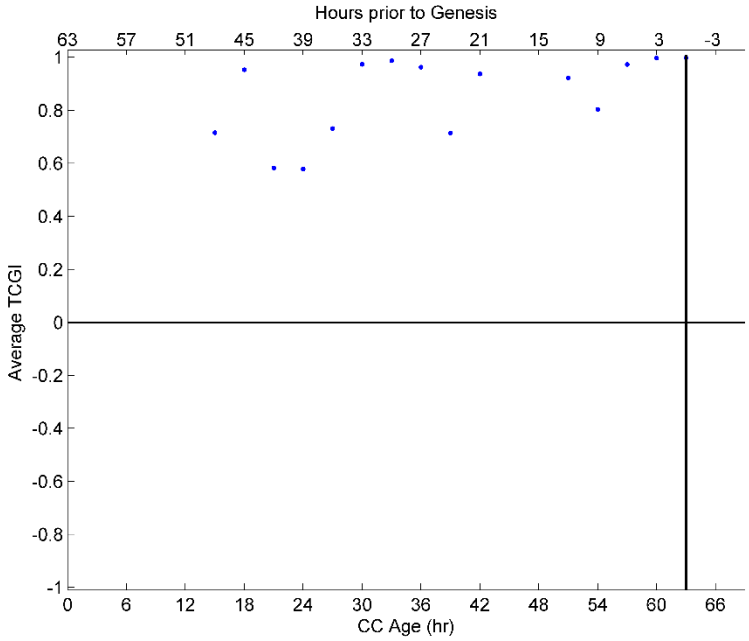


Figure 32. Average TCGI values for Hurricane Olga (2001) for the neural network simulation.

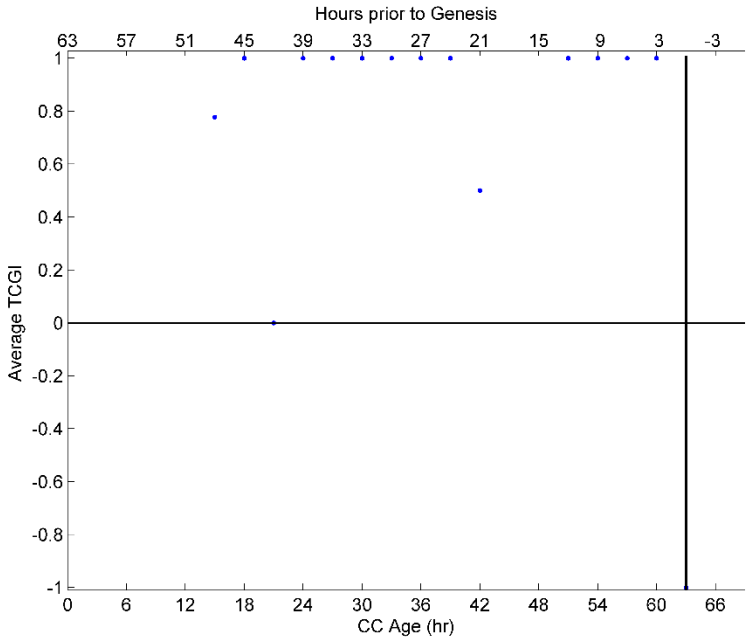


Figure 33. Average TCGI values for Hurricane Olga (2001) for the support vector machine simulation.

In summary, the CCs of Hurricane Olga (2001) are accurately identified as developing by all evaluated classifiers. The CART simulation identifies the CC at 0Z November 23 as unfavorable for development while the remaining simulations identify all CCs as favorable. This is attributed to the simplicity of the CART algorithm but the overall results indicate that when using our techniques, the pre-Olga CCs are developing with high confidence.

6.3 Hurricane Michelle (2001)

The pre-Michelle CC originated at 6Z October 27 with a geometric center at (6.71°N, 76.63°W). This CC was produced from a tropical wave moving westward from the coast of Africa, an increase in shower activity, and the formation of a low pressure area near the Nicaragua coast (Beven, 2002). It was not until 18Z October 29 that an Air Force Reserve Hurricane Hunter aircraft identified the system as a TC. Figure 34 displays the CC at genesis from our CC dataset and from the HURSAT dataset after applying our BT threshold of 250K. The difference in the defined centers is visible by the shift of the CC in the image.

(a) (b)

Figure 34. Hurricane Michelle at 18Z October 29 from (a) our CC dataset and (b) the HURSAT data after applying our brightness temperature threshold.

A series of 0-48 hour forecasts were issued for the pre-Michelle CCs using the CART, neural network, and SVM simulations. The TCGI values and average TCGI values for each forecast hour for the CART simulations are displayed in Figure 35 and Figure 36, respectively. This classifier does a good job at identifying the developing stage of the system. The first forecast to identify the storm as developing is the 36 hour forecast for the CC observed at 18Z October 27 which was exactly 48 hours prior to its genesis even though some forecast suggest unfavorable conditions. When considering the average TCGI values, all TCGI values are above 0.66 which are highly favorable conditions with the exception of the CC observation at 18Z October 27 (~0.36) which is still favorable and 15Z October 29 (-1) which is highly unfavorable.

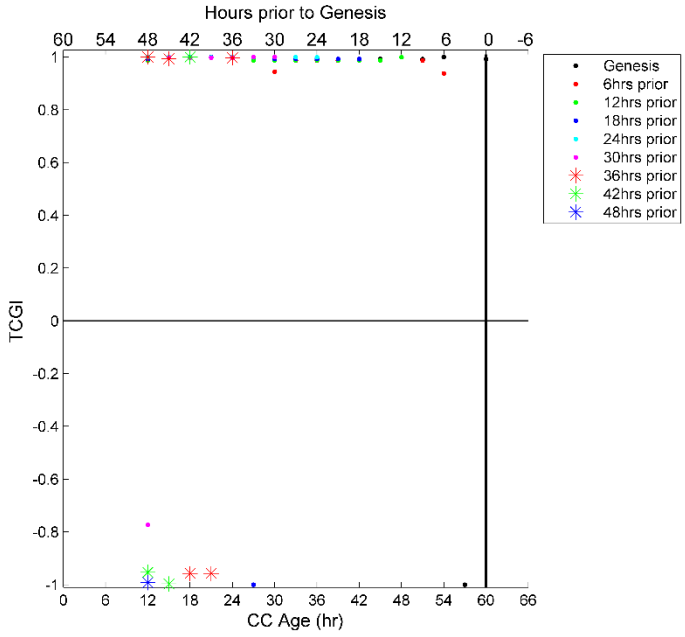


Figure 35. TCGI values for each forecast hour for Hurricane Michelle (2001) for the CART simulation.

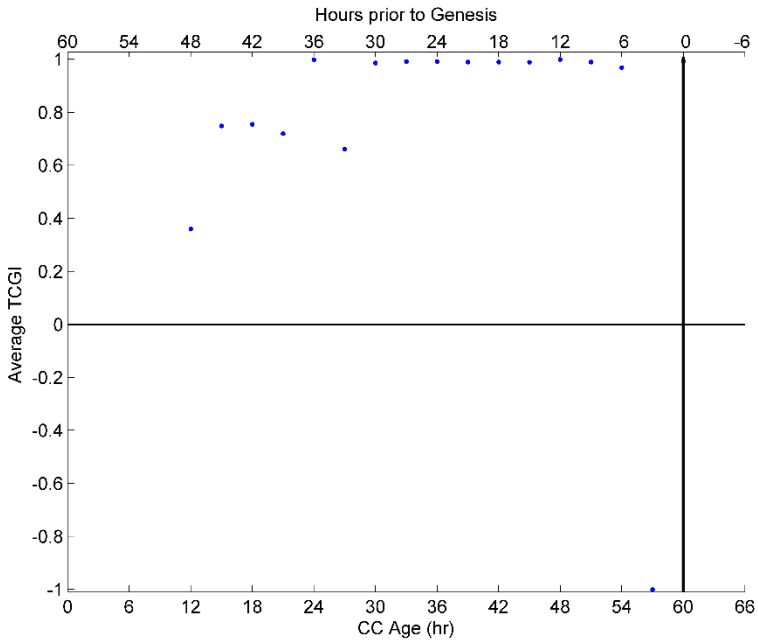


Figure 36. Average TCGI values for Hurricane Michelle (2001) for the CART simulation.

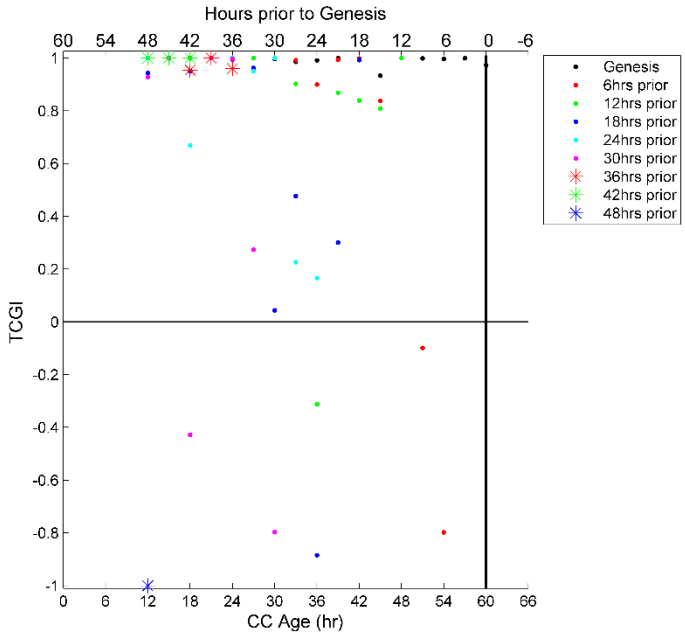


Figure 37. TCGI values for each forecast hour for Hurricane Michelle (2001) for the neural network simulation.

The TCGI values and average TCGI values for each forecast hour for the neural network simulations are displayed in Figure 37 and Figure 38, respectively. The first forecast to identify the storm as developing is the 42 hour forecast for the CC observed at 18Z October 27. When considering the average TCGI values, all TCGI values are favorable conditions even though the TCGI values oscillate between 0.17 and 1.

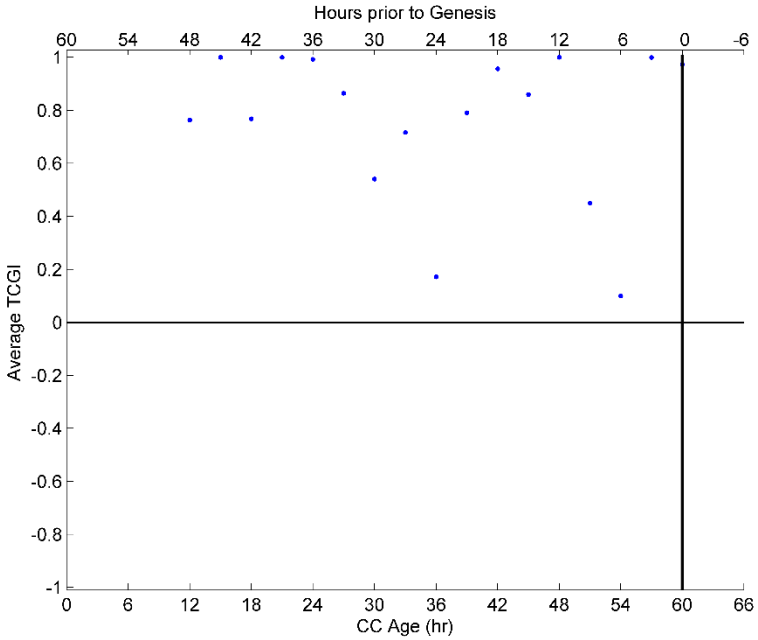


Figure 38. Average TCGI values for Hurricane Michelle (2001) for the neural network simulation.

The TCGI values and average TCGI values for each forecast hour for the SVM simulations are displayed in Figure 39 and Figure 40, respectively. The 36 hour forecast is the first forecast to suggest favorable conditions for CC development as in the CART simulation. The difference between the simulations is the SVM simulation suggest unfavorable conditions for more CC observations than the CART simulation. This is also visible with the average TCGI values that are displayed in Figure 40. The observations at 3Z October 28 (-0.14), 12Z October 28 (-0.67), 18Z October 28 (-0.2), and 12Z October 29 (-1) have unfavorable conditions for

development while observations at 21Z October 28 and 9Z October 29 have neutral conditions. This suggests that the SVM simulation suffers more than the other simulations as evidenced by the oscillations in the forecasts. The SVM simulation is a non-probabilistic classifier and does not use an optimal decision threshold as the other classifiers. Hence, some observations of this case study are similar to the non-developing CCs and are misclassified.

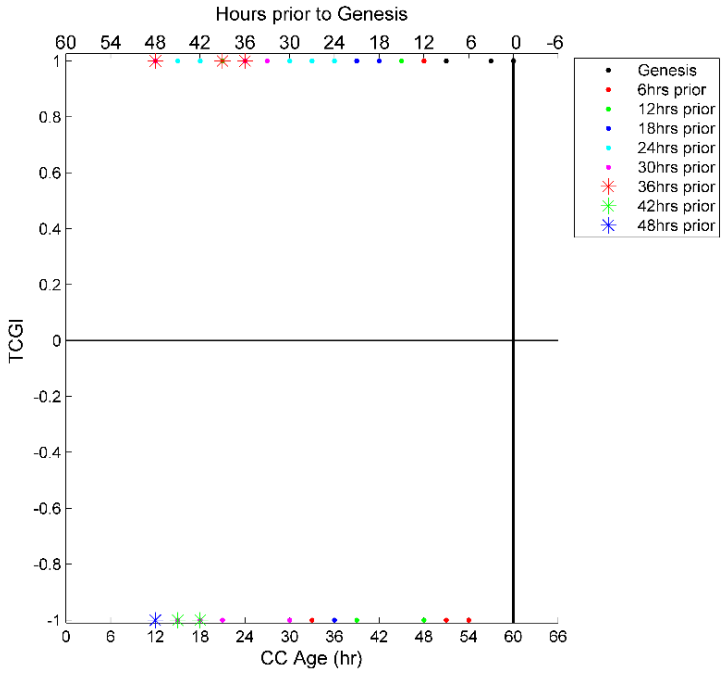


Figure 39. TCGI values for each forecast hour for Hurricane Michelle (2001) for the support vector machine simulation.

Overall, all classifiers performed well during the phases of development. When comparing the average TCGI values over all classifiers, Hurricane Michelle can be identified as a developing CC 48 hours prior to development which is at 18Z October 27.

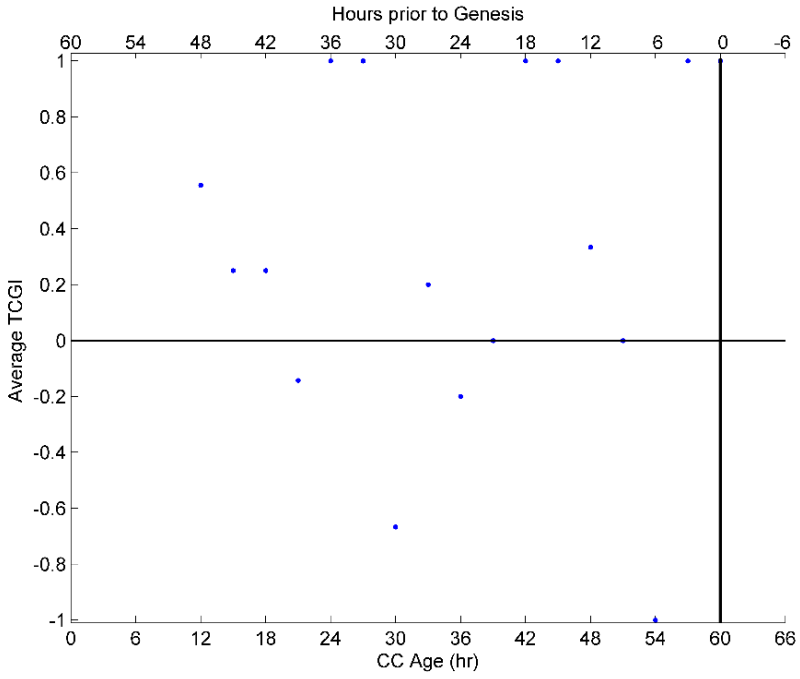


Figure 40. Average TCGI values for Hurricane Michelle (2001) for the support vector machine simulation.

6.4 Non-developing Case 100 (ND-100 2003)

This system formed off the west coast of Africa during the beginning of the 2003 North Atlantic hurricane season at 9Z June 3. It then persisted for 33 hours and dissipated at 18Z June 4. The track of this non-developing CC is displayed in Figure 41. Through simulations, the predictive features reveal that ND-100 was unfavorable for its entire lifetime and predicted no development. This is illustrated by the consistency in TCGI values for the CART simulation with TCGI values less than -0.89, the neural network with TCGI values less than -0.96, and SVM simulations with TCGI values equal to -1 in Figure 42 through Figure 44, respectively. All simulations suggest there was no chance for development.

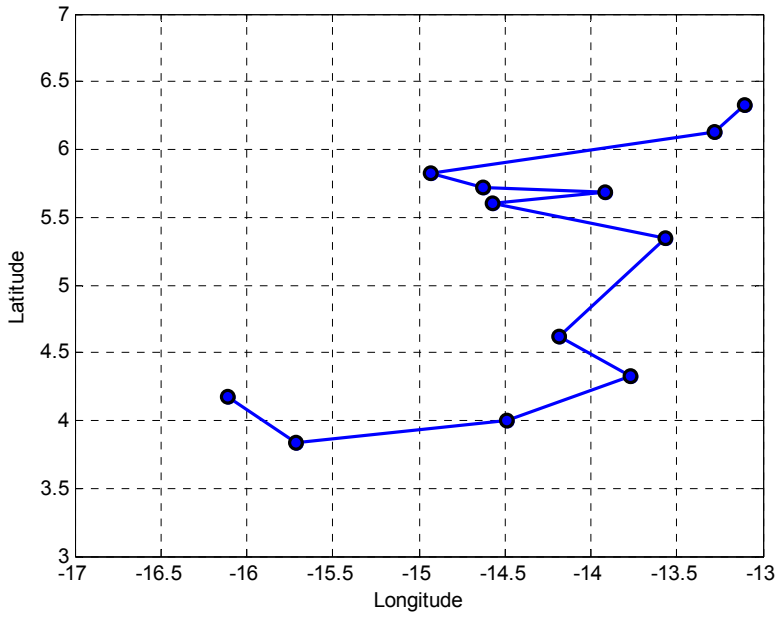


Figure 41. Irregular track of cloud cluster of ND-100 (2003).

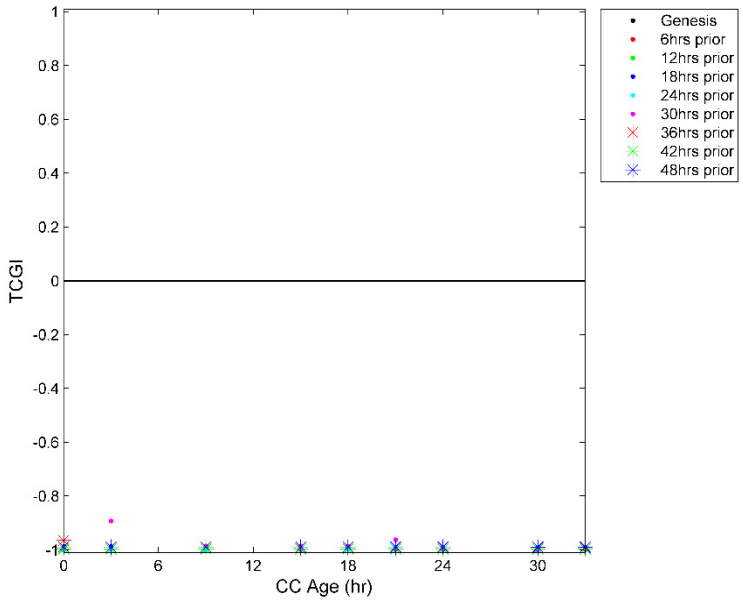


Figure 42. TCGI values for each forecast hour for ND-100 (2003) for the CART simulation.

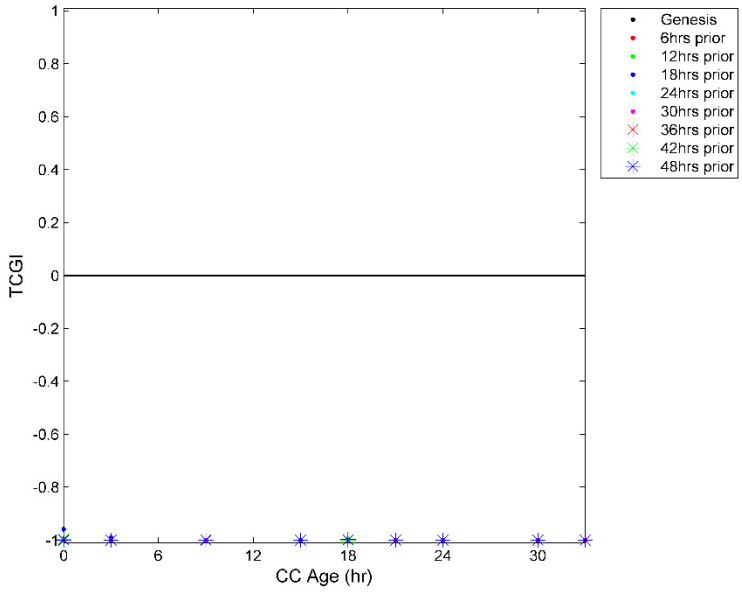


Figure 43. TCGI values for each forecast hour for ND-100 (2003) for the neural network simulation.

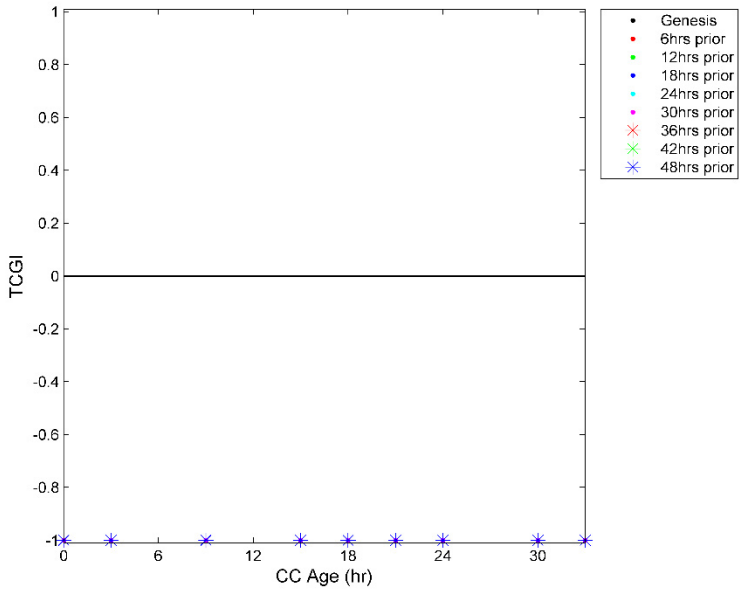


Figure 44. TCGI values for each forecast hour for ND-100 (2003) for the support vector machine simulation.

6.5 Non-developing Case 101 (ND-101 2000)

The non-developing case ND-101 (2000) formed off the northeast coast of South America during the beginning of the 2000 North Atlantic hurricane season at 21Z June 2. It then persisted for 24 hours and dissipated at 21Z June 3. The track of this non-developing CC is displayed in Figure 45. Through simulations, the predictive features reveal that ND-101 was unfavorable for its entire lifetime and predicted no development. This is illustrated by the consistency in TCGI values, which are all less than -0.96 for the CART, neural network, and SVM simulations in Figure 46 through Figure 48, respectively. Regardless of which classifier is used, the forecasts suggest there was no chance for development.

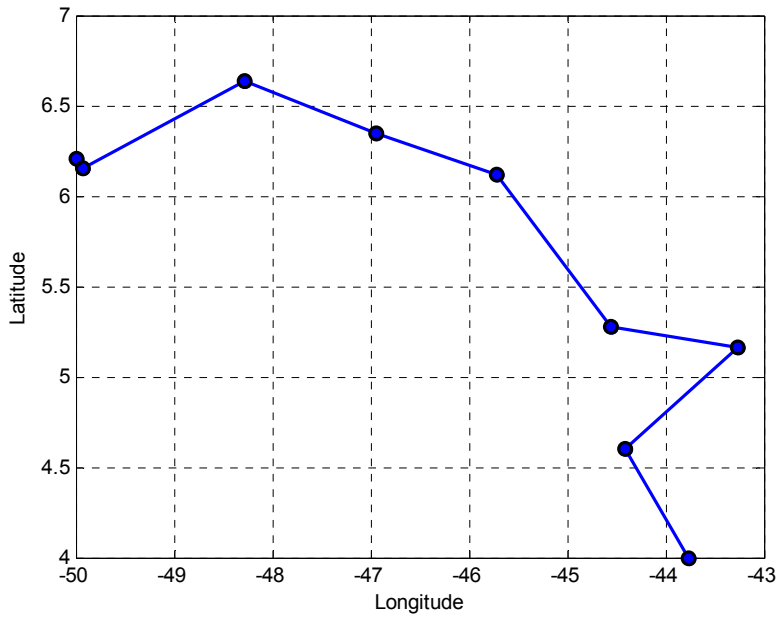


Figure 45. Irregular track of cloud cluster of ND-101 (2000).

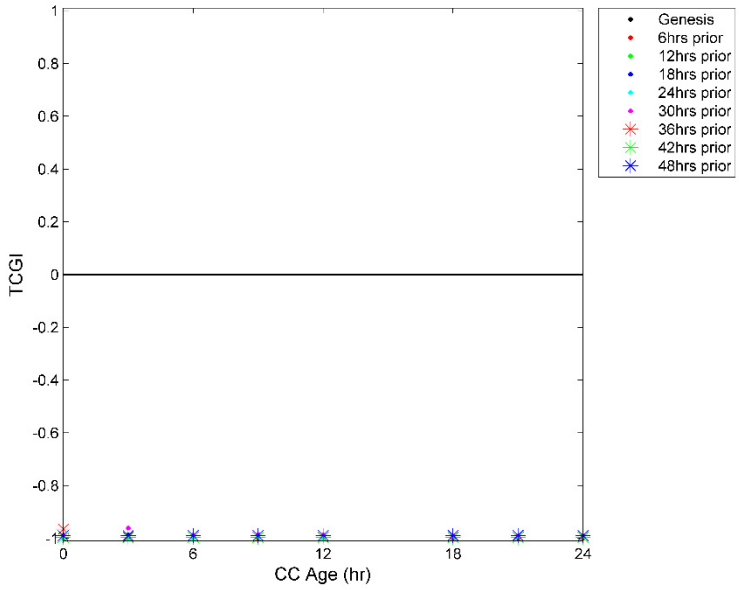


Figure 46. TCGI values for each forecast hour for ND-101 (2000) for the CART simulation.

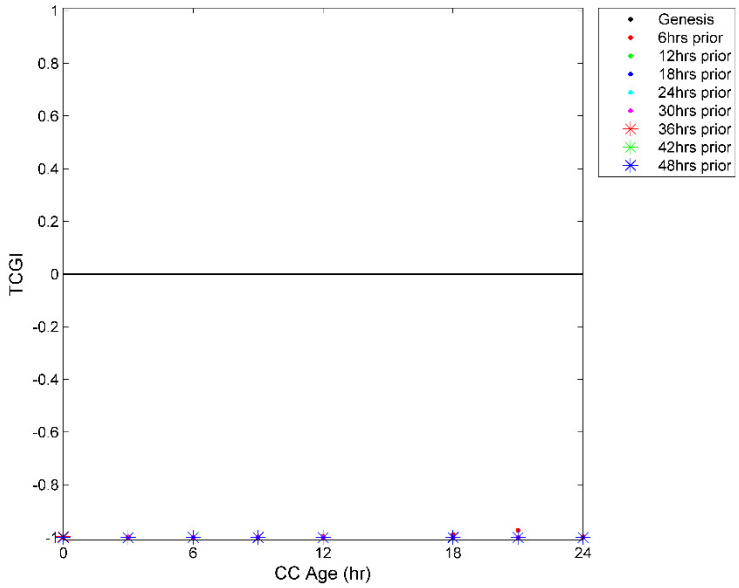


Figure 47. TCGI values for each forecast hour for ND-101 (2000) for the neural network simulation.

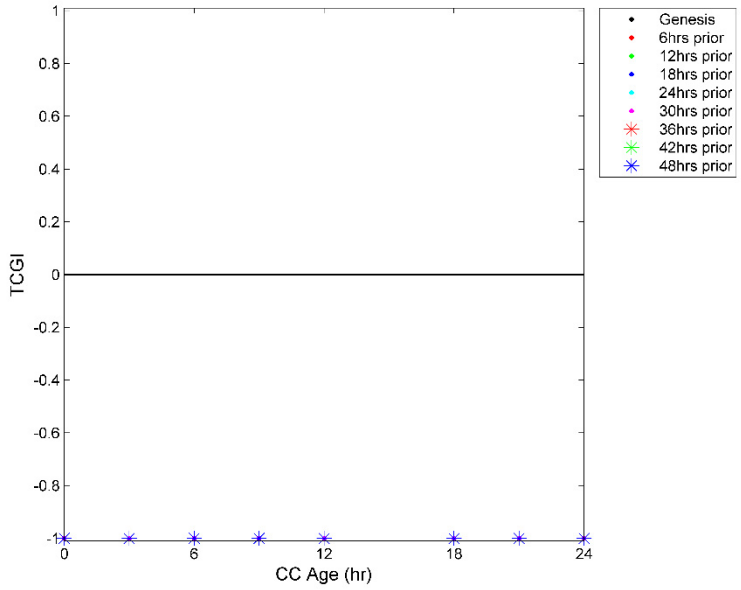


Figure 48. TCGI values for each forecast hour for ND-101 (2000) for the support vector machine simulation.

6.6 Non-developing Case 1007 (ND-1007 2002)

This system formed off the west coast of Africa during the beginning of the 2002 North Atlantic hurricane season at 18Z June 20. It then persisted for 24 hours and dissipated at 18Z June 21. The track of this non-developing CC is displayed in Figure 49. Through simulations, the predictive features reveal that ND-1007 was unfavorable for its entire lifetime and predicted no development. This is illustrated by the consistency in TCGI values for the CART, neural network, and SVM simulations in Figure 50 through Figure 52, respectively. The neural network simulation shows CC observation that are more favorable than the other simulations. Even though these observations have TCGI value higher than the minimum value of -1, they remain highly unfavorable with TCGI value less than -0.72. Regardless of which classifier is used, the forecasts suggest there was no chance for development.

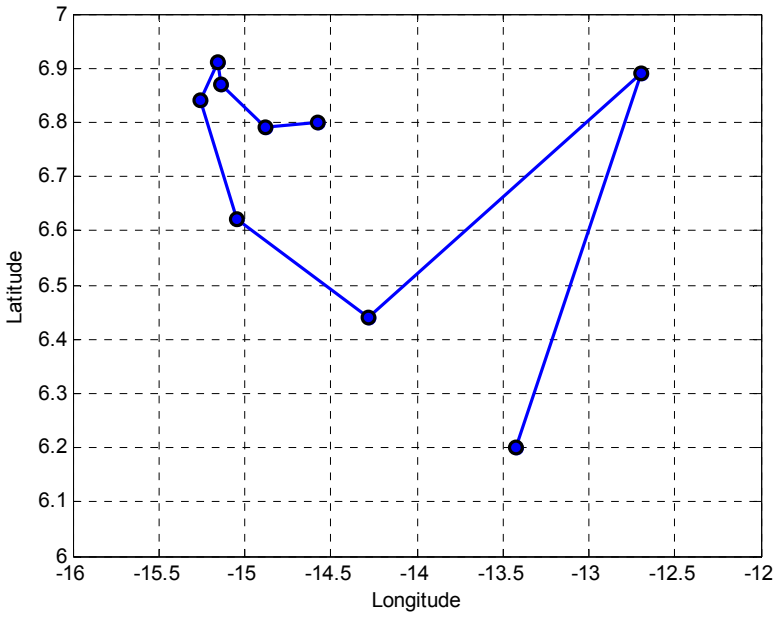


Figure 49. Irregular track of cloud cluster of ND-1007 (2002).

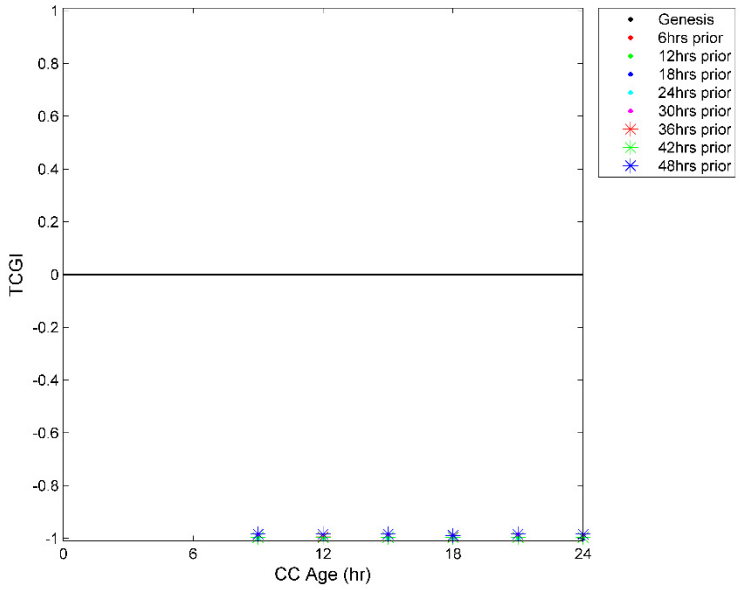


Figure 50. TCGI values for each forecast hour for ND-1007 (2002) for the CART simulation.

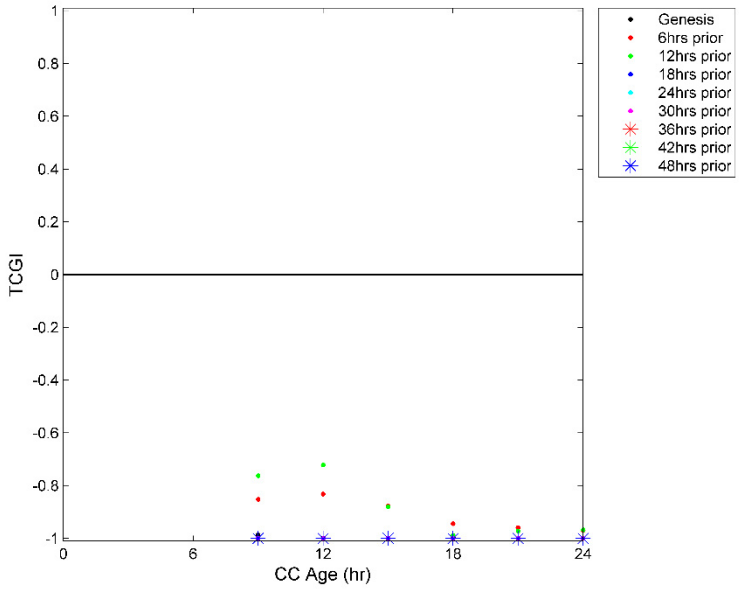


Figure 51. TCGI values for each forecast hour for ND-1007 (2002) for the neural network simulation.

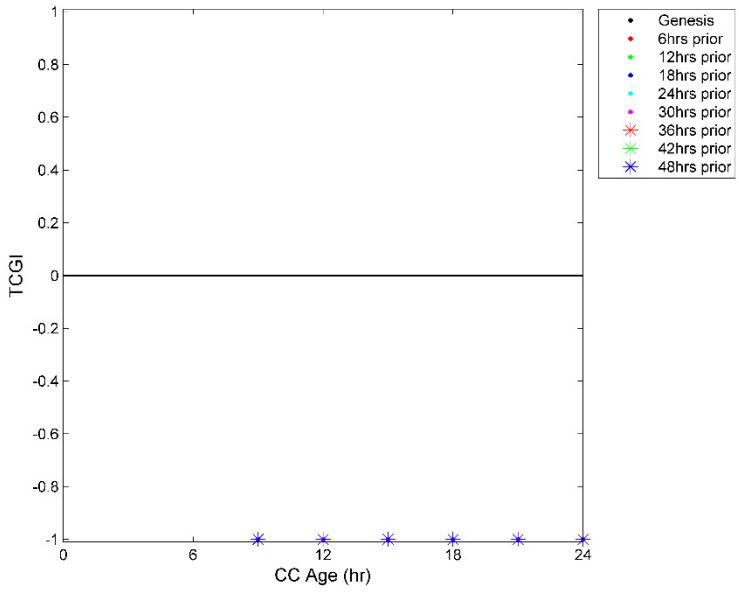


Figure 52. TCGI values for each forecast hour for ND-1007 (2002) for the support vector machine simulation.

6.7 Tropical Storm Debby (2006)

To verify the performance of our techniques, we evaluated Tropical Storm Debby from the 2006 North Atlantic hurricane season since this hurricane season is not included in the dataset. The pre-Debby CC originated at 0Z August 20 with a geometric center at (9.67°N, 8.97°W). This CC was produced from a tropical wave moving westward across the west coast of Africa and established closed circulation directly after moving offshore. The pre-Debby CC was initially identified by the Dvorak Classification method at 12Z August 21 and at 18Z it was labeled a tropical depression near Praia in the Cape Verde Islands (Franklin, 2007). Figure 34 displays the CC at genesis from our CC dataset and from the HURSAT dataset after applying our BT threshold of 250K. The difference in the defined centers is visible by the minor shift of the CC in the image.

(a) (b)

Figure 53. Tropical storm Debby at 18Z August 21 from (a) our CC dataset and (b) the HURSAT data after applying our brightness temperature threshold.

A series of 0-48 hour forecasts were issued for the pre-Debby CCs using the CART, neural network, and SVM simulations. The TCGI values and average TCGI values for each forecast hour for the CART simulations are displayed in Figure 54 and Figure 55, respectively. This classifier does a good job at identifying the developing stage of the system. The first forecast to identify the storm as developing is the 36 hour forecast for the CC observed at 0Z August 20 which was exactly 42 hours prior to its genesis even though some forecast suggest unfavorable conditions. When considering the average TCGI values, all TCGI values are above 0.99 which are highly favorable conditions with the exception of the CC observation at 0Z August 20 (~ -0.19) which is slightly unfavorable, and 3Z August 20, 6Z August 20, 12Z August 20, and 15Z August 21 (< -0.69) which are highly unfavorable.

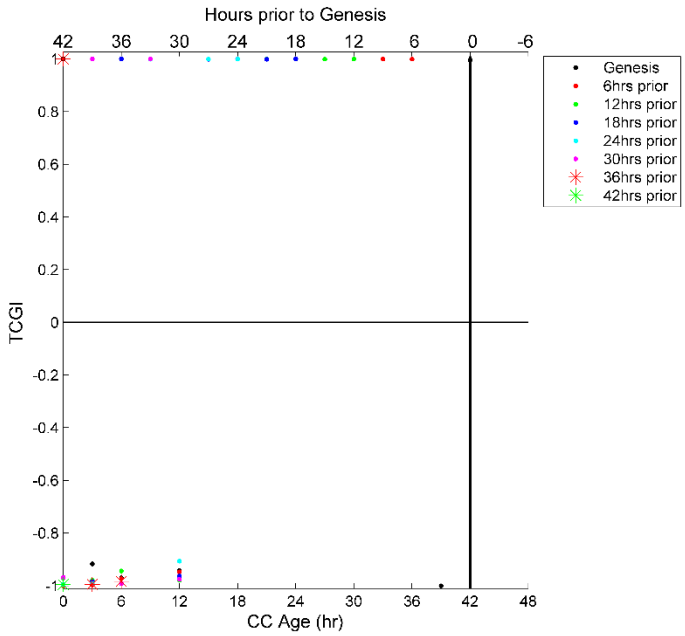


Figure 54. TCGI values for each forecast hour for Tropical Storm Debby (2006) for the CART simulation.

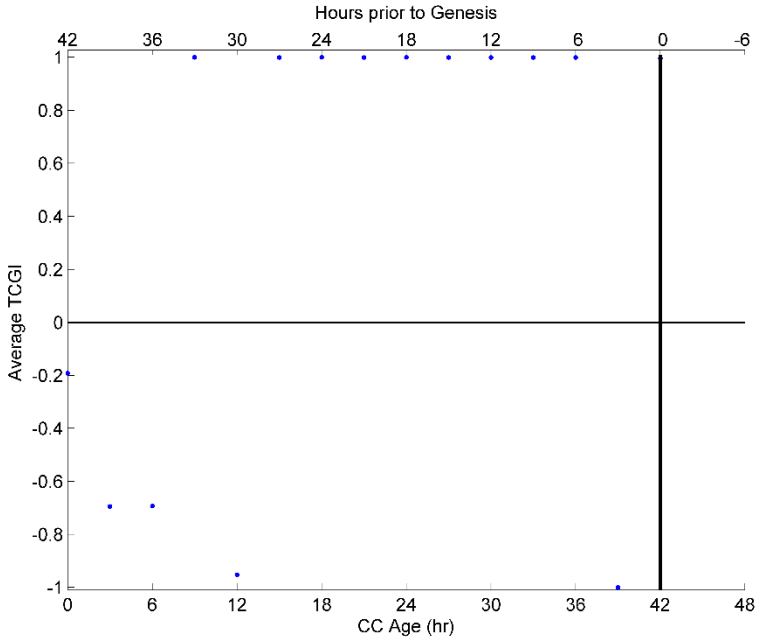


Figure 55. Average TCGI values for Tropical Storm Debby (2006) for the CART simulation.

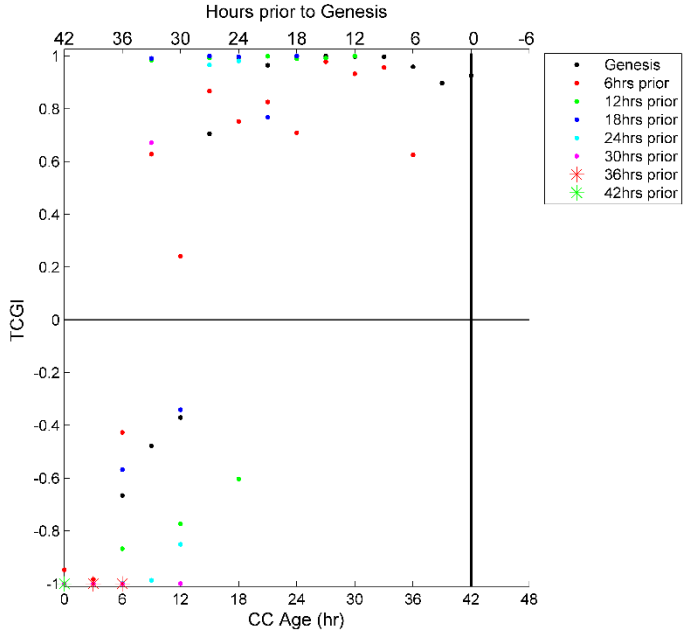


Figure 56. TCGI values for each forecast hour for Tropical Storm Debby (2006) for the neural network simulation.

The TCGI values and average TCGI values for each forecast hour for the neural network simulations are displayed in Figure 56 and Figure 57, respectively. The first forecast to identify the storm as developing is the 30 hour forecast for the CC observed at 9Z August 20. When considering the average TCGI values, all TCGI values beginning at 9Z August 20 are favorable conditions with an exception for the CC observed at 12Z August 20.

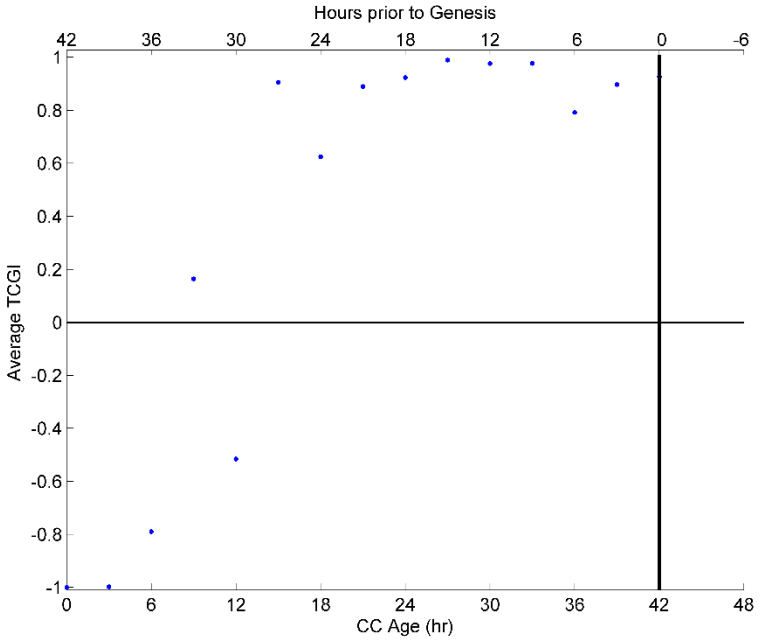


Figure 57. Average TCGI values for Tropical Storm Debby (2006) for the neural network simulation.

The TCGI values and average TCGI values for each forecast hour for the SVM simulation are displayed in Figure 58 and Figure 59, respectively. The 24 hour forecast is the first forecast to suggest favorable conditions for CC development. The SVM simulation suggest unfavorable conditions for more CC observations than the CART and neural network simulations. This is also visible with the average TCGI values that are displayed in Figure 59. The observations at 15Z August 20 (1), 21Z August 20 (0.50), 3Z August 21 (1), 6Z August 21 (1), and 9Z August 21 (1) have favorable conditions for development while observations at 12Z

August 21 have neutral conditions. This suggests that the SVM simulations suffers more than the other simulations as evidenced by the oscillations in the forecasts.

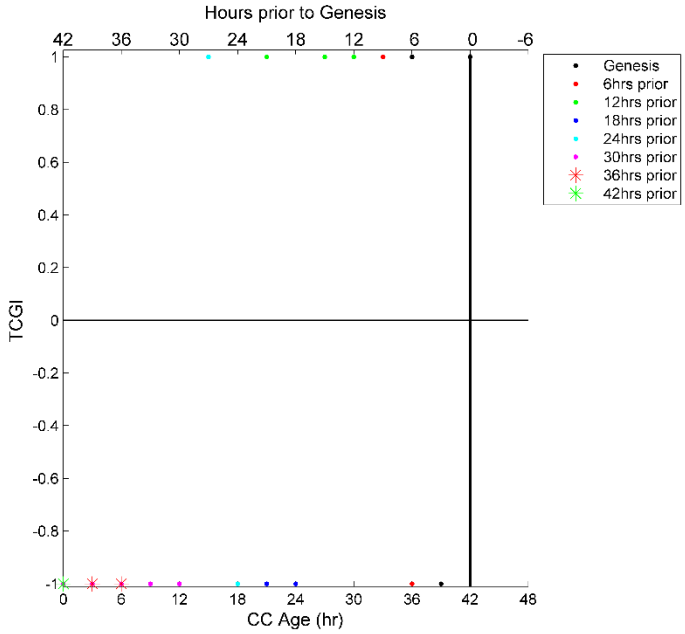


Figure 58. TCGI values for each forecast hour for Tropical Storm Debby (2006) for the support vector machine simulation.

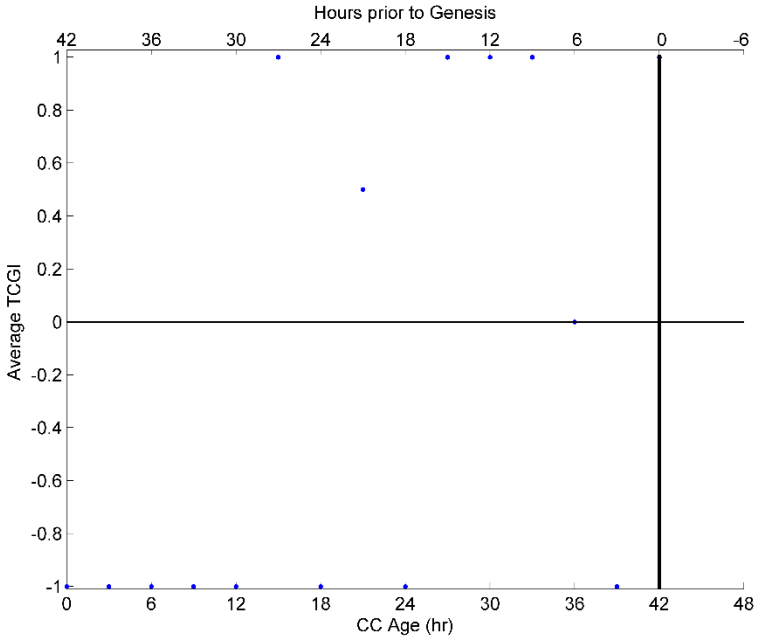


Figure 59. Average TCGI values for Tropical Storm Debby (2006) for the support vector machine simulation.

Overall, the CART and neural network classifiers performed well during the phases of development. When comparing the average TCGI values over all classifiers, Tropical Storm Debby (2006) can be identified as a developing CC 33 hours prior to development which is at 9Z August 20 which is 27 hours prior to being identified with the Dvorak classification method.

Table 15

Summary of case study characteristics

		Katrina (2005)	Olga (2001)	Michelle (2001)	Debby (2006)
Genesis Date		18Z August 23	6Z November 23	18Z October 29	18Z August 21
Hours before Genesis that TWO Conveyed CC as Developing		36	-	-	6
Earliest Detection Date	***CART***	3Z August 22	6Z November 21	18Z October 27	0Z August 20
	Neural Network	6Z August 22	6Z November 21	18Z October 27	9Z August 20
	SVM	3Z August 22	6Z November 21	18Z October 27	15Z August 20
Earliest Detection Forecast Hour	***CART***	36	42	36	36
	Neural Network	30	42	42	30
	SVM	36	42	36	24
Earliest Detection in Hours Before Genesis	***CART***	39	48	48	42
	Neural Network	36	48	48	33
	SVM	39	48	48	27

6.8 Summary

Six case studies were randomly chosen, with an exception of Hurricane Katrina (2005), from the 1999-2005 North Atlantic hurricane seasons and the Tropical Storm Debby case study was randomly chosen from the 2006 hurricane season to verify the performance of our techniques. Table 15 summarizes important characteristics from the case studies of developed TCs including the number of hours prior to genesis that the TWO indicated the CC has a

possibility of development. The TWO did not indicate the time of possible development for Hurricane Olga (2001) and Hurricane Michelle (2001). In each case, our techniques identifies developing CCs at least 27 hours prior to genesis in the 24 hour forecast in the worst case scenario. This scenario is from Tropical Storm Debby (2006) which is a weaker form of a TC than a hurricane. Therefore, its characteristics may not compare to other TCs until closer to genesis. In the best case scenario, the developing CC is identified 48 hours prior to genesis in the 42 hour forecast. These results further verify that the suggested predictive features can identify developing CCs using the aforementioned methods.

CHAPTER 7

Conclusion and Future Work

7.1 Conclusion

The formation of TCs over the North Atlantic Ocean continues to be an important research topic due to lack of scientific understanding and sparse data. The National Hurricane Center currently use forecasting models to assist in the prediction and preparedness of TCs but accurate models of TC development remain elusive. These models are initialized by satellite data and the model attempts to forecast atmospheric processes. This study suggests that the use of actual satellite observations can satisfactorily assist in providing imperative information regarding developing TCs. This research specifically focuses on identifying predictive features of developing CCs in the North Atlantic Ocean without expert subjectivity and without using forecasting models. This research topic needs attention especially since the United States is directly impacted by the activity in the North Atlantic Ocean and forecasters lack scientific understanding. Forecasters can gain valuable knowledge from feature extraction, and oversampling of satellite observations to improve forecasts and preparedness for TCs. Satellite observations have assisted in understanding atmospheric properties and examining the evolution of CCs in many studies. Therefore, this research verifies that it is beneficial to use the satellite observations to identify predictive features of developing CCs.

Identifying CCs in satellite observations is a difficult task due to the multiple definitions of a CC. Therefore, we produced a new dataset that contains eighty features of CCs in the North Atlantic Ocean that are used to identify predictive features. The most important portion of this research is objectively identifying and tracking individual CCs in order to analyze their movements and identify important characteristics that contribute to the development or non-

development of a CC into a TC. To produce this dataset, we identified individual CCs that formed above the equator and south of 40°N by examining the 1999-2005 North Atlantic hurricane seasons and using a conservative BT threshold of 250 K and size threshold of 5,000 km^2. The conservative thresholds identify a great number of CCs and can account for the complex processes and movement of CCs. This is beneficial because CCs are too unique and complex to be incorporated into existing dynamical models since CC patterns have a variety of shapes and forms that could change rapidly.

For each CC in the dataset, eighty features computed from actual satellite observations were extracted and can be categorized as location, shape, statistical, or image features. The contributed dataset containing all features for each CC will be available to the community to further research on tropical cyclogenesis. These features are evaluated to determine which predictive features contribute to the development of a TC. To trace the evolution of each CC, a simple area overlap method is incorporated with the use of the maximum and minimum scaled overlap parameter. The evolution of each CC is contained in a multivariate time series where each time series is labeled as a developing or non-developing CC. A set of all time steps of all CCs were evaluated using the sequential forward selection method as specified in Chapter 4 to identify possible predictive features. The results of this method identified the following twelve features as predictors: latitude of maximum genesis productivity, distance to nearest TC, average latitude of the minimum BT, eccentricity, average SST, BT in which 5% of the CC pixels are colder, percentage of CC pixels less than 195 K (-78.15°C), standard deviation of BT in rings with a radius of 200 and 550 km from the geometric center, standard deviation of BT, binary entropy, and normalized moment of inertia. The selected predictive features can make an indication regarding developing CCs. The three location features provides vital information

regarding the location of the CC, the shape feature indicates that there is a good separation between the eccentricity of developing and non-developing CCs, the six statistical features suggest that attributes of the BT of the CCs are separable, and the two image features show that the CC's resistance to rotational changes and the content in a binary image of a CC contain some imperative information about the CC. Therefore, all feature types (location, shape, statistical, and image) are required to successfully identify developing CCs from solely gridded satellite data.

The number of non-developing CCs outnumbers the number of developing CCs. Therefore, this problem is considered an imbalanced classification problem. To address this problem, we contributed a unique oversampling technique called the Selective Clustering based Oversampling Technique (SCOT) that uses a combination of local outlier factors to identify outliers, agglomerative clustering to best fit the data, and it explores the neighborhood of informative developing CC observations to reduce the risk of overfitting when generating synthetic samples. The SCOT is a technique that can be applied to identifying rare events. Therefore, SCOT was applied to the data using only the identified predictive features and an age parameter. SCOT generated synthetic developing CC samples to make the class distribution of the developing and non-developing CCs approximately equal which reduces the bias of the non-developing CCs when using a standard classifier. The G-Mean and HSS results for each forecast verify that forecasters can identify a developing CC analyzing the identified predictive features up to 48 hours before the CC actually develops.

Our proposed oversampling technique SCOT can assist in the identification of rare events in many applications. Such applications include the identification of cancer, suicidal behavior, tornadoes, and other rare events that directly impact society. Through this research we

discovered that unlike other oversampling techniques, SCOT eliminates user defined parameters, identifies hard to learn minority samples better, produces synthetic samples to better define the decision boundary, and generates synthetic samples in the area of the minority class to avoid overlapping of classes which, in all, lowers the risk of overfitting. These benefits and the applicability of SCOT are confirmed by a comparison with state-of-the-art techniques when using twelve real-world datasets that contain less than ten percent minority samples. The results from this comparison are found in Appendix B which were originally presented in Lacewell and Homaifar (2015). Overall, this technique could allow researchers to gain additional knowledge on events that do not occur frequently.

Our approach for identifying predictive features of developing CCs demonstrates predictive skill for 0 - 48 hours prior to development and current methods have satisfactory predictive skills approximately 24 hours prior to genesis. The case studies presented in the preceding chapter also verify the ability of our approach by identifying developing CCs 27 – 48 hours prior to genesis using the 24 – 42 hour forecasts. Overall, the results demonstrate that our approach could potentially improve weather prediction and provide advance notice of a developing CC. Having warnings in advance can avoid or reduce the risk of damages and allow emergency responders and the affected community enough time to respond appropriately.

7.2 Future Work

In future, rare event problems such as detecting cancer in medical patients, detecting suicidal behavior, detecting tornadoes, and etc., can benefit from the application of our proposed oversampling technique SCOT. Furthermore, our proposed techniques could be implemented in real time by using real-time gridded satellite data and the identified predictors. Real-time implementation could keep society updated on TC development by using solely observations that

have occurred and without depending on numerical models which are typically only initiated by satellite data. A forecaster can simply monitor real time satellite imagery, extract the suggested predictors, and identify developing CCs using a standard classifier. In addition to real-time implementation, this work could also be applied to other ocean basins to determine whether the identified predictors are dependent on processes in the North Atlantic Ocean or whether they apply to other basins. Overall, this research could assist in improving weather prediction.

References

Arnaud, Y., Desbois, M., & Maizi, J. (1992). Automatic Tracking and Characterization of African Convective Systems on Meteosat Pictures. *Journal of Applied Meteorology*, *31*(5), 443–453. http://doi.org/10.1175/1520-0450(1992)031<0443:ATACOA>2.0.CO;2

Avila, L. A. (2001, December). Tropical Cyclone Report: Hurricane Olga. Retrieved from http://www.nhc.noaa.gov/data/tcr/AL172001_Olga.pdf

Barua, S., Islam, M. M., Yao, X., & Murase, K. (2014). MWMOTE–Majority Weighted Minority Oversampling Technique for Imbalanced Data Set Learning. *IEEE Transactions on Knowledge and Data Engineering*, *26*(2), 405–425. http://doi.org/10.1109/TKDE.2012.232

Batista, G. E. A. P. A., Prati, R. C., & Monard, M. C. (2004). A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data. *SIGKDD Explor. Newsl.*, *6*(1), 20–29. http://doi.org/10.1145/1007730.1007735

Beale, M. H., Hagan, M. T., & Demuth, H. B. (2013, September). MATLAB Neural Network Toolbox User's Guide, R2013b.

Bekkar, M., & Alitouche, T. A. (2013). Imbalanced Data Learning Approaches Review. *International Journal of Data Mining & Knowledge Management Process*, *3*(4), 15–33. http://doi.org/10.5121/ijdkp.2013.3402

Bekkar, M., Djemaa, H. K., & Alitouche, T. A. (2013). Evaluation Measures for Models Assessment over Imbalanced Data Sets. *Journal of Information Engineering and Applications*, *3*(10), 27–38.

Berry, G. J., & Thorncroft, C. (2005). Case Study of an Intense African Easterly Wave. *Monthly Weather Review*, *133*(4), 752–766. http://doi.org/10.1175/MWR2884.1

Beven, J. (2002, January). Tropical Cyclone Report: Hurricane Michelle. Retrieved from http://www.nhc.noaa.gov/data/tcr/AL152001_Michelle.pdf

Bissonnette, V. L. (2011). Critical Values of the Wilcoxon Signed Ranks Test. Retrieved February 17, 2014, from http://facultyweb.berry.edu/vbissonnette/tables/wilcox_t.pdf

Blachnik, M. (2009). Comparison of Various Feature Selection Methods in Application to Prototype Best Rules. In M. Kurzynski & M. Wozniak (Eds.), *Computer Recognition Systems 3* (pp. 257–264). Springer Berlin Heidelberg. Retrieved from http://link.springer.com/chapter/10.1007/978-3-540-93905-4_31

Boer, E. R., & Ramanathan, V. (1997). Lagrangian approach for deriving cloud characteristics from satellite observations and its implications to cloud parameterization. *Journal of Geophysical Research: Atmospheres*, *102*(D17), 21383–21399. http://doi.org/10.1029/97JD00930

Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). LOF: Identifying Density-based Local Outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data* (pp. 93–104). New York, NY, USA: ACM. http://doi.org/10.1145/342009.335388

Brown, G., Pocock, A., Zhao, M.-J., & Luján, M. (2012). Conditional Likelihood Maximisation: A Unifying Framework for Information Theoretic Feature Selection. *J. Mach. Learn. Res.*, *13*, 27–66.

Bunkhumpornpat, C., Sinapiromsaran, K., & Lursinsap, C. (2012). DBSMOTE: Density-Based Synthetic Minority Over-sampling TEchnique. *Applied Intelligence*, *36*(3), 664–684. http://doi.org/http://0-dx.doi.org.sheba.ncat.edu/10.1007/s10489-011-0287-y

Cao, H., Li, X.-L., Woon, D. Y.-K., & Ng, S.-K. (2013). Integrated Oversampling for Imbalanced Time Series Classification. *IEEE Transactions on Knowledge and Data Engineering*, *25*(12), 2809–2822. http://doi.org/10.1109/TKDE.2013.37

Carvalho, L. M. V., & Jones, C. (2001). A Satellite Method to Identify Structural Properties of Mesoscale Convective Systems Based on the Maximum Spatial Correlation Tracking Technique (MASCOTTE). *Journal of Applied Meteorology*, *40*(10), 1683–1701. http://doi.org/10.1175/1520-0450(2001)040<1683:ASMTIS>2.0.CO;2

Carvalho, L. M. V., Lavallée, D., & Jones, C. (2002). Multifractal properties of evolving convective systems over tropical South America. *Geophysical Research Letters*, *29*(15), 33–1–33–4. http://doi.org/10.1029/2001GL014276

Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, *40*(1), 16–28. http://doi.org/10.1016/j.compeleceng.2013.11.024

Chang, C.-P. (1970). Westward Propagating Cloud Patterns in the Tropical Pacific as seen from Time-Composite Satellite Photographs. *Journal of the Atmospheric Sciences*, *27*(1), 133–138. http://doi.org/10.1175/1520-0469(1970)027<0133:WPCPIT>2.0.CO;2

Chawla, N. V. (2005). Data Mining for Imbalanced Datasets: An Overview. In O. Maimon & L. Rokach (Eds.), *Data Mining and Knowledge Discovery Handbook* (pp. 853–867). Springer US. Retrieved from http://link.springer.com/chapter/10.1007/0-387-25465-X_40

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, *16*(1), 321–357.

Chen, S., Guo, G., & Chen, L. (2010). A New Over-Sampling Method Based on Cluster Ensembles. In *2010 IEEE 24th International Conference on Advanced Information*

Networking and Applications Workshops (WAINA) (pp. 599–604). http://doi.org/10.1109/WAINA.2010.40

Chen, S., He, H., & Garcia, E. A. (2010). RAMOBoost: Ranked Minority Oversampling in Boosting. *IEEE Transactions on Neural Networks*, *21*(10), 1624–1642. http://doi.org/10.1109/TNN.2010.2066988

Chen, Y., Li, Y., Cheng, X.-Q., & Guo, L. (2006). Survey and Taxonomy of Feature Selection Algorithms in Intrusion Detection System. In H. Lipmaa, M. Yung, & D. Lin (Eds.), *Information Security and Cryptology* (pp. 153–167). Springer Berlin Heidelberg. Retrieved from http://link.springer.com/chapter/10.1007/11937807_13

Dare, R. A., & McBride, J. L. (2011). The Threshold Sea Surface Temperature Condition for Tropical Cyclogenesis. *Journal of Climate*, *24*(17), 4570–4576. http://doi.org/10.1175/JCLI-D-10-05006.1

Demsar, J. (2006). Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, *7*, 1–30.

Dias, N. S., Kamrunnahar, M., Mendes, P. M., Schiff, S. J., & Correia, J. H. (2010). Feature selection on movement imagery discrimination and attention detection. *Medical and Biological Engineering and Computing*, *48*(4), 331–41. http://doi.org/http://0-dx.doi.org.sheba.ncat.edu/10.1007/s11517-010-0578-1

Dolce, C. (2013, June). Top 10 Costliest Hurricanes. Retrieved March 25, 2014, from http://www.weather.com/news/weather-hurricanes/ten-most-costly-hurricanes-20130524?pageno=10

Doukim, C. A., Dargham, J. A., & Chekima, A. (2010). Finding the number of hidden neurons for an MLP neural network using coarse to fine search technique. In *2010 10th*

International Conference on Information Sciences Signal Processing and their Applications (ISSPA) (pp. 606–609). http://doi.org/10.1109/ISSPA.2010.5605430

Eleyan, A., & Demirel, H. (2011). Co-occurrence matrix and its statistical features as a new approach for face recognition. *Turkish Journal of Electrical Engineering & Computer Sciences*, *19*(1), 97–107.

Exelis Visual Information Solutions. (2014). Products & Services. Retrieved January 8, 2014, from http://www.exelisvis.com/ProductsServices/IDL.aspx

Feidas, H., & Cartalis, C. (2005). Application of an automated cloud-tracking algorithm on satellite imagery for tracking and monitoring small mesoscale convective cloud systems. *International Journal of Remote Sensing*, *26*(8), 1677–1698. http://doi.org/10.1080/01431160512331338023

Fernández, A., García, S., & Herrera, F. (2011). Addressing the Classification with Imbalanced Data: Open Problems and New Challenges on Class Distribution. In E. Corchado, M. Kurzyński, & M. Woźniak (Eds.), *Hybrid Artificial Intelligent Systems* (pp. 1–10). Springer Berlin Heidelberg. Retrieved from http://link.springer.com/chapter/10.1007/978-3-642-21219-2_1

Franklin, J. L. (2007, January). Tropical Cyclone Report: Tropical Storm Debby. Retrieved from http://www.nhc.noaa.gov/data/tcr/AL052006_Debby.pdf

Furey, T. S., Cristianini, N., Duffy, N., Bednarski, D. W., Schummer, M., & Haussler, D. (2000). Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, *16*(10), 906–914. http://doi.org/10.1093/bioinformatics/16.10.906

Futyan, J. M., & Del Genio, A. D. (2007). Deep Convective System Evolution over Africa and the Tropical Atlantic. *Journal of Climate*, *20*(20), 5041–5060. http://doi.org/10.1175/JCLI4297.1

García, V., Sánchez, J. S., Mollineda, R. A., Alejo, R., & Sotoca, J. M. (2007). The class imbalance problem in pattern classification and learning (pp. 283–291). Presented at the Congreso Español de Informática 2007, Zaragoza. Retrieved from http://marmota.dlsi.uji.es/WebBIB/papers/2007/1_GarciaTamida2007.pdf

Gavrishchaka, V. V., & Ganguli, S. B. (2001). Support vector machine as an efficient tool for high-dimensional data processing: Application to substorm forecasting. *Journal of Geophysical Research: Space Physics*, *106*(A12), 29911–29914. http://doi.org/10.1029/2001JA900118

Gray, W. M. (1968). GLOBAL VIEW OF THE ORIGIN OF TROPICAL DISTURBANCES AND STORMS. *Monthly Weather Review*, *96*(10), 669–700. http://doi.org/10.1175/1520-0493(1968)096<0669:GVOTOO>2.0.CO;2

Grazzini, J., Bereziat, D., & Herlin, I. (2001). Analysis of cloudy structures evolution on meteorological satellite acquisitions. In *2001 International Conference on Image Processing, 2001. Proceedings* (Vol. 3, pp. 760–763 vol.3). http://doi.org/10.1109/ICIP.2001.958230

Guo, Z., Dai, X., & Wu, J. (2008). An Automatic Tracking Approach for Monitoring Moving Targets from Meteorological Satellite Image Sequence Based on Point-Pattern Matching. In *Congress on Image and Signal Processing, 2008. CISP '08* (Vol. 4, pp. 150–155). http://doi.org/10.1109/CISP.2008.296

Guyon, I., & Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. *J. Mach. Learn. Res.*, *3*, 1157–1182.

Han, H., Wang, W.-Y., & Mao, B.-H. (2005). Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. In D.-S. Huang, X.-P. Zhang, & G.-B. Huang (Eds.), *Advances in Intelligent Computing* (pp. 878–887). Springer Berlin Heidelberg. Retrieved from http://0-link.springer.com.sheba.ncat.edu/chapter/10.1007/11538059_91

Han, M., & Liu, X. (2013). Feature selection techniques with class separability for multivariate time series. *Neurocomputing*, *110*, 29–34. http://doi.org/10.1016/j.neucom.2012.12.006

Hart, P. (1968). The condensed nearest neighbor rule (Corresp.). *IEEE Transactions on Information Theory*, *14*(3), 515–516. http://doi.org/10.1109/TIT.1968.1054155

Hastie, T., Tibshirani, R., & Friedman, J. (2009). Unsupervised Learning. In *The Elements of Statistical Learning* (pp. 485–585). Springer New York. Retrieved from http://0-link.springer.com.sheba.ncat.edu/chapter/10.1007/978-0-387-84858-7_14

He, H., Bai, Y., Garcia, E. A., & Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *IEEE International Joint Conference on Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence)* (pp. 1322–1328). http://doi.org/10.1109/IJCNN.2008.4633969

He, H., & Garcia, E. A. (2009). Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, *21*(9), 1263–1284. http://doi.org/10.1109/TKDE.2008.239

Hennon, C. C. (2003). *Investigating Probabilistic Forecasting of Tropical Cyclogenesis Over the North Atlantic Using Linear and Non-Linear Classifiers*. The Ohio State University.

Retrieved from https://etd.ohiolink.edu/ap/10?0::NO:10:P10_ACCESSION_NUM:osu1047237423

Hennon, C. C. (2008, July). Tropical Meteorology: A First Course. Retrieved from http://atmoschang.webs.com/tropical_MET TB.pdf

Hennon, C. C., Helms, C. N., Knapp, K. R., & Bowen, A. R. (2011). An Objective Algorithm for Detecting and Tracking Tropical Cloud Clusters: Implications for Tropical Cyclogenesis Prediction. *Journal of Atmospheric and Oceanic Technology*, *28*(8), 1007–1018. http://doi.org/10.1175/2010JTECHA1522.1

Hennon, C. C., & Hobgood, J. S. (2003). Forecasting Tropical Cyclogenesis over the Atlantic Basin Using Large-Scale Data. *Monthly Weather Review*, *131*(12), 2927–2940. http://doi.org/10.1175/1520-0493(2003)131<2927:FTCOTA>2.0.CO;2

Hennon, C. C., Marzban, C., & Hobgood, J. S. (2005). Improving Tropical Cyclogenesis Statistical Model Forecasts through the Application of a Neural Network Classifier. *Weather and Forecasting*, *20*(6), 1073–1083. http://doi.org/10.1175/WAF890.1

Hopsch, S. B., Thorncroft, C. D., & Tyle, K. R. (2010). Analysis of African Easterly Wave Structures and Their Role in Influencing Tropical Cyclogenesis. *Monthly Weather Review*, *138*(4), 1399–1419. http://doi.org/10.1175/2009MWR2760.1

Houze, R. A. (2010). Clouds in Tropical Cyclones. *Monthly Weather Review*, *138*(2), 293–344. http://doi.org/10.1175/2009MWR2989.1

Hu, S., Liang, Y., Ma, L., & He, Y. (2009). MSMOTE: Improving Classification Performance When Training Data is Imbalanced. In *Second International Workshop on Computer Science and Engineering, 2009. WCSE '09* (Vol. 2, pp. 13–17). http://doi.org/10.1109/WCSE.2009.756

Japkowicz, N. (2000). Learning from Imbalanced Data Sets: A Comparison of Various Strategies. In *Proceedings of Learning from Imbalanced Data Sets, Papers from the AAAI Workshop* (pp. 10–15).

Karsoliya, S. (2012). Approximating Number of Hidden layer neurons in Multiple Hidden Layer BPNN Architecture. *International Journal of Engineering Trends and Technology*, *3*(6), 714–717.

Kekre, H. B., Thepade, S. D., Sarode, anuja K., & Suryawanshi, V. (2010). Image Retrieval using Texture Features extracted from GLCM, LBG and KPE. *International Journal of Computer Theory and Engineering*, 695–700. http://doi.org/10.7763/IJCTE.2010.V2.227

Kerns, B. W., & Chen, S. S. (2013). Cloud Clusters and Tropical Cyclogenesis: Developing and Nondeveloping Systems and Their Large-Scale Environment. *Monthly Weather Review*, *141*(1), 192–210. http://doi.org/10.1175/MWR-D-11-00239.1

Knabb, R. D., Rhome, J. R., & Brown, D. P. (2005, December). Tropical Cyclone Report: Hurricane Katrina. Retrieved from http://www.nhc.noaa.gov/pdf/TCR-AL122005_Katrina.pdf

Knapp, K. R., Ansari, S., Bain, C. L., Bourassa, M. A., Dickinson, M. J., Funk, C., … Magnusdottir, G. (2011). Globally Gridded Satellite Observations for Climate Studies. *Bulletin of the American Meteorological Society*, *92*(7), 893–907. http://doi.org/10.1175/2011BAMS3039.1

Knapp, K. R., & Kossin, J. P. (2007). New global tropical cyclone data set from ISCCP B1 geostationary satellite observations. *Journal of Applied Remote Sensing*, *1*(1), 013505–013505–6. http://doi.org/10.1117/1.2712816

Kubat, M., & Matwin, S. (1997). Addressing the Curse of Imbalanced Training Sets: One-Sided Selection. In *In Proceedings of the Fourteenth International Conference on Machine Learning* (pp. 179–186). Morgan Kaufmann.

Lacewell, C., Homaifar, A., & Lin, Y.-L. (2013). Tracing the origins and propagation of pre-tropical storm Debby (2006) mesoscale convective systems using pattern recognition and image fusion. *Meteorology & Atmospheric Physics*, *119*(1-2), 43–58. http://doi.org/10.1007/s00703-012-0214-8

Lacewell, C. W., & Homaifar, A. (2015). SCOT: Selective Clustering based Oversampling Technique. *Data Mining and Knowledge Discovery*, submitted.

Laurikkala, J. (2001). Improving Identification of Difficult Small Classes by Balancing Class Distribution. In *Proceedings of the 8th Conference on AI in Medicine in Europe: Artificial Intelligence Medicine* (pp. 63–66). London, UK, UK: Springer-Verlag. Retrieved from http://dl.acm.org/citation.cfm?id=648155.757340

Lee, C. S. (1989). Observational Analysis of Tropical Cyclogenesis in the Western North Pacific. Part I: Structural Evolution of Cloud Clusters. *Journal of the Atmospheric Sciences*, *46*(16), 2580–2598. http://doi.org/10.1175/1520-0469(1989)046<2580:OAOTCI>2.0.CO;2

Lewis, M. (2012). *Applied Statistics for Economists*. Routledge.

Lin, Y.-L. (2007). *Mesoscale Dynamics*. Cambridge University Press.

Lin, Y.-L., Liu, L., Tang, G., Spinks, J., & Jones, W. (2013). Origin of the pre-tropical storm Debby (2006) African easterly wave-mesoscale convective system. *Meteorology and Atmospheric Physics*, *120*(3-4), 123–144. http://doi.org/10.1007/s00703-013-0248-6

Liu, L., Sun, X., Chen, F., Zhao, S., & Gao, T. (2011). Cloud Classification Based on Structure Features of Infrared Images. *Journal of Atmospheric and Oceanic Technology*, *28*(3), 410–417. http://doi.org/10.1175/2010JTECHA1385.1

Luengo, J., Fernández, A., García, S., & Herrera, F. (2011). Addressing data complexity for imbalanced data sets: analysis of SMOTE-based oversampling and evolutionary undersampling. *Soft Computing*, *15*(10), 1909–1936. http://doi.org/10.1007/s00500-010-0625-8

Machado, L. A. T., Rossow, W. B., Guedes, R. L., & Walker, A. W. (1998). Life Cycle Variations of Mesoscale Convective Systems over the Americas. *Monthly Weather Review*, *126*(6), 1630–1654. http://doi.org/10.1175/1520-0493(1998)126<1630:LCVOMC>2.0.CO;2

Mandal, A. K., Pal, S., De, A. K., & Mitra, S. (2005). Novel approach to identify good tracer clouds from a sequence of satellite images. *IEEE Transactions on Geoscience and Remote Sensing*, *43*(4), 813–818. http://doi.org/10.1109/TGRS.2005.843324

Mapes, B. E., & Houze, R. A. (1993). Cloud Clusters and Superclusters over the Oceanic Warm Pool. *Monthly Weather Review*, *121*(5), 1398–1416. http://doi.org/10.1175/1520-0493(1993)121<1398:CCASOT>2.0.CO;2

Marill, T., & Green, D. M. (1963). On the effectiveness of receptors in recognition systems. *IEEE Transactions on Information Theory*, *9*(1), 11–17. http://doi.org/10.1109/TIT.1963.1057810

MathWorks Incorporated. (2005). Learning MATLAB. Retrieved from http://www.mathworks.com/academia/student_version/learnmatlab_sp3.pdf

MathWorks Incorporated. (2014). Selecting Features for Classifying High-dimensional Data. Retrieved from http://www.mathworks.com/help/stats/examples/selecting-features-for-classifying-high-dimensional-data.html

McTaggart-Cowan, R., Deane, G. D., Bosart, L. F., Davis, C. A., & Galarneau, T. J. (2008). Climatology of Tropical Cyclogenesis in the North Atlantic (1948–2004). *Monthly Weather Review*, *136*(4), 1284–1304. http://doi.org/10.1175/2007MWR2245.1

Mingqiang, Y., Kidiyo, K., & Joseph, R. (2008). A Survey of Shape Feature Extraction Techniques. In P.-Y. Yin (Ed.), *Pattern Recognition Techniques, Technology and Applications*. InTech. Retrieved from http://www.intechopen.com/books/pattern_recognition_techniques_technology_and_applications/a_survey_of_shape_feature_extraction_techniques

Montgomery, M. T., Davis, C., Dunkerton, T., Wang, Z., Velden, C., Torn, R., … Boothe, M. A. (2012). The Pre-Depression Investigation of Cloud-Systems in the Tropics (PREDICT) Experiment: Scientific Basis, New Analysis Tools, and Some First Results. *Bulletin of the American Meteorological Society*, *93*(2), 153–172. http://doi.org/10.1175/BAMS-D-11-00046.1

Narsky, I., & Porter, F. C. (2013). Decision Trees. In *Statistical Analysis Techniques in Particle Physics* (pp. 307–329). Wiley-VCH Verlag GmbH & Co. KGaA. Retrieved from http://onlinelibrary.wiley.com/doi/10.1002/9783527677320.ch14/summary

National Oceanic and Atmospheric Administration. (2009, July 24). Technical Summary of the National Hurricane. Retrieved from http://www.nhc.noaa.gov/pdf/model_summary_20090724.pdf

Peng, M. S., Fu, B., Li, T., & Stevens, D. E. (2012). Developing versus Nondeveloping Disturbances for Tropical Cyclone Formation. Part I: North Atlantic*. *Monthly Weather Review*, *140*(4), 1047–1066. http://doi.org/10.1175/2011MWR3617.1

Piñeros, M. F., Ritchie, E. A., & Tyo, J. S. (2010). Detecting Tropical Cyclone Genesis From Remotely Sensed Infrared Image Data. *IEEE Geoscience and Remote Sensing Letters*, *7*(4), 826–830. http://doi.org/10.1109/LGRS.2010.2048694

Pohjalainen, J., Räsänen, O., & Kadioglu, S. (2013). Feature selection methods and their combinations in high-dimensional classification of speaker likability, intelligibility and personality traits. *Computer Speech & Language*. http://doi.org/10.1016/j.csl.2013.11.004

Räsänen, O., & Pohjalainen, J. (2013). Random subset feature selection in automatic recognition of developmental disorders, affective states, and level of conflict from speech. In *INTERSPEECH 2013* (pp. 210–214). Lyon, France.

Reed, R. J., Norquist, D. C., & Recker, E. E. (1977). The Structure and Properties of African Wave Disturbances as Observed During Phase III of GATE. *Monthly Weather Review*, *105*(3), 317–333. http://doi.org/10.1175/1520-0493(1977)105<0317:TSAPOA>2.0.CO;2

Reynolds, R. W., Smith, T. M., Liu, C., Chelton, D. B., Casey, K. S., & Schlax, M. G. (2007). Daily High-Resolution-Blended Analyses for Sea Surface Temperature. *Journal of Climate*, *20*(22), 5473–5496. http://doi.org/10.1175/2007JCLI1824.1

Saeys, Y., Inza, I., & Larrañaga, P. (2007). A review of feature selection techniques in. *Bioinformatics*, *23*(19), 2507–2517. http://doi.org/10.1093/bioinformatics/btm344

Shahamiri, S. R., & Binti Salim, S. S. (2014). Real-time frequency-based noise-robust Automatic Speech Recognition using Multi-Nets Artificial Neural Networks: A multi-views multi-

learners approach. *Neurocomputing*, *129*, 199–207. http://doi.org/10.1016/j.neucom.2013.09.040

Shao, Y., & Lunetta, R. S. (2012). Comparison of support vector machine, neural network, and CART algorithms for the land-cover classification using limited training data points. *ISPRS Journal of Photogrammetry and Remote Sensing*, *70*, 78–87. http://doi.org/10.1016/j.isprsjprs.2012.04.001

Sheela, K. G., & Deepa, S. N. (2013). Review on Methods to Fix Number of Hidden Neurons in Neural Networks. *Mathematical Problems in Engineering*, *2013*, e425740. http://doi.org/10.1155/2013/425740

Shen, B.-W., Tao, W.-K., Lau, W. K., & Atlas, R. (2010). Predicting tropical cyclogenesis with a global mesoscale model: Hierarchical multiscale interactions during the formation of tropical cyclone Nargis (2008). *Journal of Geophysical Research: Atmospheres*, *115*(D14), D14102. http://doi.org/10.1029/2009JD013140

Sherwood, S. C., & Wahrlich, R. (1999). Observed Evolution of Tropical Deep Convective Events and Their Environment. *Monthly Weather Review*, *127*(8), 1777–1795. http://doi.org/10.1175/1520-0493(1999)127<1777:OEOTDC>2.0.CO;2

Simpson, J., Ritchie, E., Holland, G. J., Halverson, J., & Stewart, S. (1997). Mesoscale Interactions in Tropical Cyclone Genesis. *Monthly Weather Review*, *125*(10), 2643–2661. http://doi.org/10.1175/1520-0493(1997)125<2643:MIITCG>2.0.CO;2

Song, Q., Ni, J., & Wang, G. (2013). A Fast Clustering-Based Feature Subset Selection Algorithm for High-Dimensional Data. *IEEE Transactions on Knowledge and Data Engineering*, *25*(1), 1–14. http://doi.org/10.1109/TKDE.2011.181

Sun, Y., Wong, A. K. C., & Kamel, M. S. (2009). Classification of Imbalanced Data: A Review. *International Journal of Pattern Recognition & Artificial Intelligence*, *23*(4), 687–719.

Terry, J. P. (2007). Tropical Cyclogenesis. In *Tropical Cyclones* (pp. 15–25). Springer New York. Retrieved from http://0-link.springer.com.sheba.ncat.edu/chapter/10.1007/978-0-387-71543-8_2

Vila, D. A., Machado, L. A. T., Laurent, H., & Velasco, I. (2008). Forecast and Tracking the Evolution of Cloud Clusters (ForTraCC) Using Satellite Infrared Imagery: Methodology and Validation. *Weather and Forecasting*, *23*(2), 233–245. http://doi.org/10.1175/2007WAF2006121.1

Weiss, G. M. (2013). Foundations of Imbalanced Learning. In H. He & Yunqian (Eds.), *Imbalanced Learning* (pp. 13–41). John Wiley & Sons, Inc. Retrieved from http://onlinelibrary.wiley.com/doi/10.1002/9781118646106.ch2/summary

Weiss, S. M., & Indurkhya, N. (1997). *Predictive Data Mining: A Practical Guide* (1 edition). San Francisco: Morgan Kaufmann.

Whitney, A. W. (1971). A Direct Method of Nonparametric Measurement Selection. *IEEE Transactions on Computers*, *C-20*(9), 1100–1103. http://doi.org/10.1109/T-C.1971.223410

Wilder, M. J. (2011). *Automatic Target Recognition: Statistical Feature Selection of Non-Gaussian Distributed Target Classes*.

Wilks, D. S. (2005). *Statistical Methods in Atmospheric Sciences, Volume 91 : An Introduction*. Burlington, MA, USA: Academic Press. Retrieved from http://site.ebrary.com/lib/alltitles/docDetail.action?docID=10138343

Williams, M., & Houze, R. A. (1987). Satellite-Observed Characteristics of Winter Monsoon Cloud Clusters. *Monthly Weather Review*, *115*(2), 505–519. http://doi.org/10.1175/1520-0493(1987)115<0505:SOCOWM>2.0.CO;2

Wilson, D. L. (1972). Asymptotic Properties of Nearest Neighbor Rules Using Edited Data. *IEEE Transactions on Systems, Man and Cybernetics*, *SMC-2*(3), 408–421. http://doi.org/10.1109/TSMC.1972.4309137

Yang, Y., Lin, H., Guo, Z., Fang, Z., & Jiang, J. (2004). Automatic tracking and characterization of multiple moving clouds in satellite images. In *2004 IEEE International Conference on Systems, Man and Cybernetics* (Vol. 4, pp. 3088–3093 vol.4). http://doi.org/10.1109/ICSMC.2004.1400813

Yoon, H., Yang, K., & Shahabi, C. (2005). Feature subset selection and feature ranking for multivariate time series. *IEEE Transactions on Knowledge and Data Engineering*, *17*(9), 1186–1198. http://doi.org/10.1109/TKDE.2005.144

Zhang, S., & Zhou, Q. (2012). New feature extraction algorithm for satellite image non-linear small objects. In *2012 IEEE Symposium on Electrical Electronics Engineering (EEESYM)* (pp. 626–629). http://doi.org/10.1109/EEESym.2012.6258736

Appendix A

This Appendix provides additional information regarding equations and descriptions of possible predictive features for cloud clusters. The following features are categorized as location, shape, statistical, or image features.

A.1 Location Features

The following features are related to the location of each CC.

- **Latitude of maximum genesis productivity (*ALAT17*):** Genesis productivity is at its maximum at ~17°N (Kerns & Chen, 2013). Therefore, the following is used as a CC feature:

$$ALAT17 = |glat - 17|$$

  where $glat$ is the latitude coordinate of the geometric center.

- **Distance to nearest TC (d_{TC}):** This feature is equivalent to the distance (in km) between the geometric center of the CC and the nearest TC origin.

- **Front edge position:** This feature provides the position of the front edge of the CC. This position is very subjective (Arnaud et al., 1992; Feidas & Cartalis, 2005); hence, it is not included in our CC feature dataset .

- **Geometric center (*glon*, *glat*):** The geometric center, also known as centroid or center of gravity, is the calculated center based on the shape of the CC. It is calculated using the following equation:

$$glon = \frac{1}{N}\sum_{i=1}^{N} X_i \qquad glat = \frac{1}{N}\sum_{i=1}^{N} Y_i$$

  where N denotes the number of points in the CC, and X_i and Y_i denote the coordinates of the points in the CC (Carvalho & Jones, 2001; Feidas & Cartalis, 2005; Hennon et al., 2011; Mingqiang, Kidiyo, & Joseph, 2008).

- **Minimum BT location (*mlon*, *mlat*):** This feature indicates the average location of the minimum BT of the CC. It is calculated as follows:

$$mlon = \frac{1}{N_B}\sum_{i=1}^{N_B} X_{Bi} \quad mlat = \frac{1}{N_B}\sum_{i=1}^{N_B} Y_{Bi}$$

where N_B denotes the number of points equal to the minimum BT, and X_{Bi} and Y_{Bi} denote the coordinates of the minimum BT points in the CC.

- **Scaled Coriolis (*SC*):** This feature is defined by:

$$SC = 2\omega \times 10^4 sin\phi$$

where ϕ denotes the latitude in degrees and ω indicates the angular rotation of the Earth which is equivalent to $7.29 \times 10^{-5}\ s^{-1}$ (Hennon & Hobgood, 2003).

- **Weighted center (*wlon*, *wlat*):** The weighted center is equivalent to the center coordinates of the CC when bias to the BTs. It is defined as follows:

$$wlon = \frac{\sum_{i=1}^{N_P} X_i BT_i}{\sum_{i=1}^{N_P} BT_i} \qquad wlat = \frac{\sum_{i=1}^{N_P} Y_i BT_i}{\sum_{i=1}^{N_P} BT_i}$$

where N_P denotes the number of pixels in the CC, BT_i denotes the BT, and X_i and Y_i denote the coordinates of the CC pixels (Arnaud et al., 1992; Carvalho & Jones, 2001; Hennon et al., 2011).

A.2 Shape Features

These features provide additional information regarding the shape of each CC.

- **Area (*A*):** The area of the CC is either in pixels or in km^2 (Arnaud et al., 1992; Carvalho & Jones, 2001; Feidas & Cartalis, 2005; Hennon et al., 2011; Kerns & Chen, 2013; Liu, Sun, Chen, Zhao, & Gao, 2011; Yang, Lin, Guo, Fang, & Jiang, 2004). If in pixels, the feature is the same as the pixel count feature.

- **Axis inclination:** Axis inclination is the clockwise angle between the first eigenvector (λ_1), also known as the major axis, of the CC and the x-axis (east-west direction) (Arnaud et al., 1992).
- **Compactness (Com):** The compactness indicates the similarity between the shape and a circle. It is very similar to roundness and is defined by:

$$C = \frac{N}{P^2}$$

where N denotes the number of pixels in the CC and P denotes the perimeter of the CC.

- **Contour of CC:** The outline of the CC is considered its contour (Mingqiang et al., 2008).
- **Eccentricity (Ecc):** Eccentricity is the ratio of the major axis length to the length of the minor axis.

$$Ecc = \frac{\lambda_2}{\lambda_1}$$

where λ_2 is the second eigenvector. High (low) E_{CC} indicates a circular (linear) CC (Carvalho & Jones, 2001; Feidas & Cartalis, 2005; Mingqiang et al., 2008). When using the contour of a CC to calculate $\lambda_1$and λ_2, the following equations are used:

$$\lambda_1 = \frac{1}{2}\left[c_{xx} + c_{yy} + \sqrt{\left(c_{xx} + c_{yy}\right)^2 - 4\left(c_{xx}c_{yy} - c_{xy}^2\right)}\right]$$

$$\lambda_2 = \frac{1}{2}\left[c_{xx} + c_{yy} - \sqrt{\left(c_{xx} + c_{yy}\right)^2 - 4\left(c_{xx}c_{yy} - c_{xy}^2\right)}\right]$$

where

$$c_{xx} = \frac{1}{N}\sum_{i=0}^{N-1}(x_i - X_C)^2$$

$$c_{xy} = c_{yx} = \frac{1}{N}\sum_{i=0}^{N-1}(x_i - X_C)(y_i - Y_C)$$

$$c_{yy} = \frac{1}{N}\sum_{i=0}^{N-1}(y_i - Y_C)^2$$

- **Ellipse Variance (E_{va}):** A mapping error of a shape to fit an ellipse with the same covariance matrix is called the ellipse variance (Mingqiang et al., 2008). This feature is defined by

$$E_{va} = \frac{\sigma'_R}{\mu'_R}$$

where we assume

$$V_i = \begin{pmatrix} x_i - glon \\ y_i - glat \end{pmatrix}$$

$$C_{ellipse} = \begin{pmatrix} c_{xx} & c_{xy} \\ c_{yx} & c_{yy} \end{pmatrix}$$

$$d'_i = \sqrt{V_i^T \cdot C_{ellipse}^{-1} \cdot V_i}$$

$$\mu'_R = \frac{1}{N}\sum_{i=1}^{N-1} d'_i$$

$$\sigma'_R = \sqrt{\frac{1}{N}\sum_{i=1}^{N-1}(d'_i - \mu'_R)^2}$$

- **Estimated radius (R_{Est}):** This feature is the estimated radius that the CC would have if it were a circle with the same area.

$$ER = \sqrt{\frac{A}{\pi}}$$

where A is the area of the CC in km^2.

- **Maximum and Minimum radius (R_{Max} and R_{Min}):** This is equivalent to the maximum and minimum radius of the CC.
- **Perimeter (P):** This feature indicates the perimeter of the CC (Carvalho & Jones, 2001).
- **Pixel count:** Pixel count (area in pixels) is equivalent to the number of pixels in each CC (Arnaud et al., 1992; Hennon et al., 2011).
- **First eigenvector length (λ_1):** This feature indicates the maximum length of the first principle axis of the CC (Arnaud et al., 1992; Feidas & Cartalis, 2005; Mingqiang et al., 2008).
- **Second eigenvector length (λ_2):** This feature indicates the maximum length of the second principle axis of the CC (Arnaud et al., 1992; Feidas & Cartalis, 2005; Mingqiang et al., 2008).
- **Protraction Ratio (G):** This feature is equivalent to the ratio of the CC's longitude coordinate range to the CC's latitude range of coordinates. It is defined as follows:

$$G = \frac{x_{max} - x_{min}}{y_{max} - y_{min}}$$

where x_{min} and y_{min} are the minimum coordinate values of the CC boundary and x_{max} and y_{max} are the maximum coordinate values. A round CC has a protraction ratio of 1 (Yang et al., 2004).

- **Roundness (Ro):** The roundness, also called the circularity ratio, indicates the similarity between the shape and a circle.

$$Ro = \frac{A}{P^2}$$

where A denotes the area of the CC in km^2 and P denotes the perimeter of the CC (Mingqiang et al., 2008; Yang et al., 2004).

A.3 Statistical Features

These features are calculated from the data of each CC.

- **Average BT (BT_{avg}):** The average BT of the CC (Carvalho & Jones, 2001; Feidas & Cartalis, 2005; Hennon et al., 2011; Liu et al., 2011).
- **Average SST (SST_{avg}):** The average SST coinciding with the CC location (Dare & McBride, 2011).
- **BT Kurtosis (BT_{kurt}):** The BT kurtosis measures the peakness of the CC distribution. This feature is defined by

$$BT_{kurt} = \frac{1}{N_P}\sum_{i=1}^{N_P}\frac{(BT_i - \mu)^4}{\sigma^4}$$

where N_P denotes the number of pixels in the CC, BT_i denotes the BT at each pixel of the CC, μ is the average BT, and σ is the standard deviation of the CC.

- **BT Skewness (BT_{skew}):** The BT skewness measures the asymmetry of the CC distribution. This feature is defined by

$$BT_{skew} = \frac{1}{N_P}\sum_{i=1}^{N_P}\frac{(BT_i - \mu)^3}{\sigma^3}$$

where N_P denotes the number of pixels in the CC, BT_i denotes the BT at each pixel of the CC, μ is the average BT, and σ is the standard deviation of the CC. A positive (negative) BT_{skew} denotes a right (left) skewed distribution.

- **5th percentile ($BT_{5\%}$):** 5% of the CC pixels are colder than this BT (Hennon et al., 2011).
- **10th percentile ($BT_{10\%}$):** 10% of the CC pixels are colder than this BT (Hennon et al., 2011).
- **Fractional convective area (Fc):** The fractional convective area represents the percentage of the CC's area that is less than or equal to 210 K. It is defined as follows:

$$Fc = 100\frac{A_{TC}}{A}$$

where A_{TC} denotes the area in a CC whose $T_B \leq 210\ K$ and A represents the area of the CC (Carvalho & Jones, 2001).

- **BT percentage (BTP_{XXX}):** The percentage of CC pixels which are less than or equal to a specified BT. Typically the specified BT is equivalent to 195K, 205K, 215K, 225K, and 235K; therefore, this produces five features. These features are defined by

$$BTP_{XXX} = 100\frac{N_{BT}}{N}$$

where N denotes the number of pixels in the CC, and N_{BT} denotes the number of pixels that are less than or equal to a specified BT (Hennon et al., 2011).

- **Minimum BT (BT_{min}):** This feature is equivalent to the minimum BT of the CC (Carvalho & Jones, 2001; Hennon et al., 2011).
- **Ring average of BT ($RingBT_{avgXX}$):** These twelve features are equivalent to the average BT in rings with a radius of 50 km, 100 km, 150 km, …, and 600 km from the geometric center of the CC.

- **Ring minimum of BT ($RingBT_{minXX}$):** These twelve features are equivalent to the minimum BT in rings with a radius of 50 km, 100 km, 150 km, …, and 600 km from the geometric center of the CC.
- **Ring standard deviation of BT ($RingBT_{stdXX}$):** These twelve features are equivalent to the standard deviation of BT in rings with a radius of 50 km, 100 km, 150 km, …, and 600 km from the geometric center of the CC.
- **Standard deviation of BT (BT_{std}):** This feature indicates the standard deviation of BTs of the CC (Hennon et al., 2011).
- **Variance of BT (BT_{var}):** This feature is the variance of all BTs in the CC (Carvalho & Jones, 2001).
- **Volume index (V):** This feature measures the potential of a CC producing heavy precipitation. The volume index is defined as

$$V = \sum n_i(T_i - T_0)$$

where n_i denotes the number of pixels in class i where each class i covers 0.5 K. T_i represents the BT of each pixel in class i and T_0 denotes a chosen BT threshold (Feidas & Cartalis, 2005).

A.4 Image Features

These features are related to the image of each CC.

- **Estimated Backscattering coefficient (EBC):** The estimated backscattering coefficient is an optical property of natural waters. It is defined (in dB) as the following:

$$EBC = 10\log\frac{\sum_{i=1}^{N} BT_i}{N}$$

where *N* denotes the number of pixels in the CC and BT_i denotes the BT of each pixel in the satellite image.

- **Binary Entropy (H_b):** Binary entropy reveals the information content of events in a binary image. The following equation is used to calculate the binary entropy:

$$H_b = \log \frac{N}{N_I}$$

where *N* denotes the number of pixels in the CC, N_I denotes the number of pixels in the satellite image, and $H_b \in [-\infty, 0]$ (Zhang & Zhou, 2012).

- **Contrast (Con):** This feature, along with correlation, energy, and homogeneity, are calculated from the gray level co-occurrence matrix (GLCM) of the CC satellite image. The GLCM is a square matrix *p* that describes the texture of an image *I* where each element *(i, j)* specifies the number of times gray level intensity *i* is adjacent to gray level intensity *j*. The GLCM is calculated using

$$p(i,j) = \sum_{x=1}^{N} \sum_{y=1}^{N} \begin{cases} 1 & if\ I(x,y) = i\ and\ I(x+\Delta_x, y+\Delta_y) \\ 0 & otherwise \end{cases}$$

where *N* denotes the number of pixels in the CC, and (Δ_x, Δ_y) is the offset that is sensitive to rotation and specifies the distance between adjacent pixels (Eleyan & Demirel, 2011; Kekre, Thepade, Sarode, & Suryawanshi, 2010). To achieve a degree of rotational invariance, a set of offsets are used as in the following table.

Table

Offsets with its corresponding angle

Angle	Offset
0	$(0, \Delta)$
45	$(-\Delta, \Delta)$
90	$(-\Delta, 0)$
135	$(-\Delta, -\Delta)$

The contrast feature uses the GLCM to measure the intensity contrast between neighboring pixels using the following equation:

$$Contrast = \sum_{i,j} |i - j|^2 p(i,j)$$

where *i* and *j* specifies the gray level intensities, and *p* is the GLCM which is dependent on the specified spatial relationships of the pixels. For example, four gray level co-occurrence matrices are produced if the GLCM is calculated by finding adjacent pixels at 0, 45, 90, and 135^o angles. Typically, if more than one GLCM is produced, the average contrast is used (Eleyan & Demirel, 2011; Kekre et al., 2010; MathWorks Incorporated, 2005).

- **Correlation (Cor):** Correlation is a measure of the correlation between neighboring pixels which is calculated from the GLCM using the following equation:

$$Correlation = \sum_{i,j} \frac{(i - \mu i)(j - \mu j)p(i,j)}{\sigma_i \sigma_j}$$

where $Correlation \in [-1,1]$. If more than one GLCM is produced, the average correlation is used (Eleyan & Demirel, 2011; Kekre et al., 2010; MathWorks Incorporated, 2005).

- **Energy ($\boldsymbol{E}$):** Energy, also known as angular second moment, is a measure of textural uniformity. It is the sum of the squared elements of the GLCM as indicated in the following equation:

$$Energy = \sum_{i,j} p(i,j)^2$$

If more than one GLCM is produced, the average energy is used (Eleyan & Demirel, 2011; Kekre et al., 2010; MathWorks Incorporated, 2005).

- **Estimated cloud fraction ($\boldsymbol{ECF}$):** The estimated cloud fraction is a ratio of the number of cloudy pixels to the total number of pixels in a CC image. It is defined as

$$ECF = \frac{N_{cloud}}{N_{total}}$$

where N_{cloud} denotes the number of cloudy pixels and N_{total} denotes the total number of pixels in the satellite image (Liu et al., 2011).

- **Homogeneity ($\boldsymbol{Hom}$):** This feature measures the closeness of the GLCM diagonal to the distribution of elements in the GLCM using the following equation:

$$Homogeneity = \sum_{i,j} \frac{p(i,j)}{1 + |i - j|}$$

If more than one GLCM is produced, the average correlation is used (Eleyan & Demirel, 2011; Kekre et al., 2010; MathWorks Incorporated, 2005).

- **Moments of inertia ($J_{(glon,glat)}$):** Moment of inertia is a measure of the CC's resistance to rotational changes. When calculating from a CC image, it is defined as follows:

$$J_{(C_x,C_y)} = \sum_{y=1}^{N}\sum_{x=1}^{M}\left[(x - C_x)^2 + \left(y - C_y\right)^2\right] f(x,y)$$

where *M* x *N* is the image size with centroid *(C_x, C_y)*, and f(x, y) denotes the intensity of image at location (x, y) (Arnaud et al., 1992; Zhang & Zhou, 2012).

- **Normalized Moment of Inertia (*NMI*):** This feature normalizes the moments of inertia as follows (Zhang & Zhou, 2012):

$$NMI = \frac{\sqrt{\sum_{y=1}^{N}\sum_{x=1}^{M}\left[(x - C_x)^2 + \left(y - C_y\right)^2\right] f(x,y)}}{\sum_{y=1}^{N}\sum_{x=1}^{M} f(x,y)}$$

Appendix B

This appendix provides the results for comparing the proposed oversampling technique SCOT to the state-of-the-art oversampling techniques SMOTE, Borderline SMOTE and MWMOTE. Descriptions of these techniques are provided in Chapter 3 and Lacewell and Homaifar (2015). Simulations were performed on each data set using 1) a basic decision tree classifier and 2) a support vector machine (SVM). For each simulation, we obtained the chosen data set, applied oversampling method, and then ran the selected classifier using ten-fold cross validation. More than one classifier is tested to verify that the results are not dependent on the chosen classifier. In the first set of simulations, a simple decision tree classifier was selected which uses pruning, has at least one observation per tree leaf, uses the Gini's diversity index as a splitting criterion, merges leaves that originate from the same parent node, and class probabilities are based on the class distribution. In the second set of simulations, a SVM classifier using the Gaussian radial basis function as the kernel function was selected with a default scaling factor (sigma) of one. The sequential minimal optimization is used to find the separating hyperplane and the maximum number of iterations to converge is set to 100,000.

Table B-1

Description of the datasets used to compare SMOTE, Borderline SMOTE, MWMOTE, and SCOT. The twelve real world imbalanced datasets contain a maximum of ten percent minority samples.

Dataset	Data Type	Minority Class	Majority Class	Features	Instances	Minority	Majority	Imbalance Ratio
Abalone	Multivariate	'18'	'9'	8	731	42	689	1:16
ClimateModel	Multivariate	'failure'	'success'	18	540	46	494	1:11
CoverType	Multivariate	'4'	'3'	54	38501	2747	35754	1:13
Mammography	Multivariate	'1'	'-1'	6	11183	260	10923	1:42
OCR	Multivariate	'0'	All others	64	3826	376	3447	1:9
OillSpill	Multivariate	'1'	'-1'	49	937	41	896	1:22
Ozone	Multivariate	'1'	'0'	73	2536	73	2463	1:34
Page Blocks	Multivariate	'3' '4' and '5'	All others	10	5476	231	5245	1:23
Robot Nav.	Multivariate	'Slight-Left-Turn'	All others	24	5456	328	5128	1:16
Seismic	Multivariate	'1'	'0'	18	2584	170	2414	1:14
Statlog	Multivariate	'4'	All others	36	6435	626	5809	1:9
TCC	Time Series	'1'	'0'	Vary	877	52	825	1:16

Table B-2

Comparison of performance measures when applying SMOTE, Borderline SMOTE, MWMOTE, and SCOT to twelve real world datasets for the decision tree simulation. The best results for each dataset are highlighted in bold.

Dataset	Method	Recall	Precision	F1 Measure	G-Mean	AUC
TCC	Raw	7.69231	10.52632	8.88889	27.15749	0.51786
TCC	SMOTE	87.0303	82.52874	84.71976	84.2589	0.84303
TCC	Borderline SMOTE	90.45505	89.36404	89.90623	89.34189	0.89349
TCC	MWMOTE	95.39394	94.2515	94.81928	94.78594	0.94788
TCC	SCOT	**98.91304**	**98.6747**	**98.79373**	**98.78978**	**0.9879**
Abalone	Raw	38.09524	37.2093	37.64706	60.49991	0.67088
Abalone	SMOTE	91.14659	90.75145	90.94859	90.92862	0.90929
Abalone	Borderline SMOTE	89.36464	89.24138	89.30297	89.02128	0.89022
Abalone	MWMOTE	97.37609	95.70201	96.53179	96.5071	0.96511
Abalone	SCOT	**98.41499**	**98.69942**	**98.557**	**98.55427**	**0.98554**
ClimateModel	Raw	30.43478	34.14634	32.18391	53.63893	0.62485
ClimateModel	SMOTE	93.52227	90.94488	92.21557	92.09436	0.92105
ClimateModel	Borderline SMOTE	86.20038	83.9779	85.07463	84.27297	0.84295
ClimateModel	MWMOTE	96.07438	94.5122	95.28689	95.30129	0.95304
ClimateModel	SCOT	**97.80439**	**98.19639**	**98**	**97.99109**	**0.97991**
OCR	Raw	94.68085	95.18717	94.93333	97.0497	0.97079
OCR	SMOTE	99.30374	99.36139	99.33256	99.33275	0.99333
OCR	Borderline SMOTE	99.27473	99.44783	99.36121	99.36173	0.99362
OCR	MWMOTE	**99.4772**	**99.73791**	**99.60739**	**99.60797**	**0.99608**
OCR	SCOT	99.41011	99.60597	99.50794	99.50194	0.99502
OilSpill	Raw	26.82927	45.83333	33.84615	51.41984	0.62689
OilSpill	SMOTE	97.32143	95.82418	96.567	96.53702	0.9654
OilSpill	Borderline SMOTE	95.62433	94.61457	95.11677	94.9639	0.94966

Table B-2

Cont.

OilSpill	MWMOTE	98.21029	97.77283	97.99107	97.98882	0.97989
OilSpill	SCOT	**99.66592**	**99.77703**	**99.72145**	**99.72134**	**0.99721**
Ozone	Raw	26.31579	27.77778	27.02703	50.73731	0.62069
Ozone	SMOTE	97.05122	95.51935	96.27919	96.06386	0.96069
Ozone	Borderline SMOTE	97.19189	95.07821	96.12343	94.96919	0.94995
Ozone	MWMOTE	98.32448	97.72543	98.02404	97.59617	0.97599
Ozone	SCOT	**99.67572**	**99.35354**	**99.51437**	**99.39077**	**0.99391**
PageBlocks	Raw	73.16017	77.16895	75.11111	85.12482	0.86103
PageBlocks	SMOTE	98.66463	98.17768	98.42055	98.41632	0.98417
PageBlocks	Borderline SMOTE	96.81085	96.79311	96.80198	96.73619	0.96736
PageBlocks	MWMOTE	99.0256	**99.06346**	99.04453	99.04542	0.99045
PageBlocks	SCOT	**99.23379**	99.03021	**99.1319**	**99.12084**	**0.99121**
Statlog	Raw	53.35463	55.48173	54.39739	71.3394	0.74371
Statlog	SMOTE	94.99053	93.58887	94.28449	94.23872	0.94242
Statlog	Borderline SMOTE	94.07947	93.94562	94.0125	93.74252	0.93743
Statlog	MWMOTE	**96.0241**	94.57535	95.29422	**95.25462**	**0.95258**
Statlog	SCOT	95.65946	**94.97504**	**95.31602**	95.22935	0.9523
RobotNav	Raw	95.73171	98.4326	97.06337	97.79487	0.97817
RobotNav	SMOTE	99.9025	99.88302	99.89276	99.89275	0.99893
RobotNav	Borderline SMOTE	99.57682	99.68687	99.63181	99.62264	0.99623
RobotNav	MWMOTE	99.86325	99.86325	99.86325	99.86337	0.99863
RobotNav	SCOT	**99.94235**	**99.94235**	**99.94235**	**99.94192**	**0.99942**
Mammography	Raw	55	64.41441	59.3361	73.89331	0.77138
Mammography	SMOTE	95.44081	97.9241	96.66651	96.70047	0.96709
Mammography	Borderline SMOTE	98.24577	98.14037	98.19304	98.17074	0.98171
Mammography	MWMOTE	99.16674	99.00357	99.08509	99.08439	0.99084

Table B-2

Cont.

Mammography	SCOT	**99.30178**	**99.17587**	**99.23879**	**99.23431**	**0.99234**
Seismic	Raw	14.70588	17.0068	15.77287	37.36665	0.54826
Seismic	SMOTE	92.58492	92.50828	92.54658	92.54349	0.92543
Seismic	Borderline SMOTE	93.43511	94.70019	94.0634	93.87889	0.9388
Seismic	MWMOTE	94.79124	95.14523	94.9679	94.97208	0.94972
Seismic	SCOT	**96.21027**	**97.00082**	**96.60393**	**96.59236**	**0.96593**
Covertype	Raw	85.22024	88.37297	86.76798	91.91633	0.92179
Covertype	SMOTE	99.02668	98.78906	98.90773	98.90634	0.98906
Covertype	Borderline SMOTE	98.70467	98.97457	98.83944	98.80689	0.98807
Covertype	MWMOTE	**99.40158**	98.93955	**99.17003**	**99.16771**	**0.99168**
Covertype	SCOT	99.12567	**99.05715**	99.0914	99.08596	0.99086

Table B-3

Comparison of performance measures when applying SMOTE, Borderline SMOTE, MWMOTE, and SCOT to twelve real world datasets for the support vector machine simulation. The best results for each dataset are highlighted in bold.

Dataset	Method	Recall	Precision	F1 Measure	G-Mean	AUC
TCC	Raw	15.38462	6.95652	9.58084	36.59136	0.51207
TCC	SMOTE	93.81818	85.80931	89.63521	89.02929	0.89152
TCC	Borderline SMOTE	94.11765	91.87432	92.98246	92.49946	0.92513
TCC	MWMOTE	96	97.05882	96.52651	96.54391	0.96545
TCC	SCOT	**99.63768**	**99.87893**	**99.75816**	**99.75816**	**0.99758**
Abalone	Raw	42.85714	19.56522	26.86567	61.84998	0.66058
Abalone	SMOTE	**99.70972**	90.87302	95.08651	94.7229	0.94848
Abalone	Borderline SMOTE	94.75138	89.90826	92.26631	91.74003	0.91788
Abalone	MWMOTE	98.54227	82.74174	89.95343	88.5303	0.89039
Abalone	SCOT	99.42363	**99.85528**	**99.63899**	**99.63901**	**0.99639**
ClimateModel	Raw	0	NaN	NaN	0	0.5
ClimateModel	SMOTE	97.57085	**100**	98.77049	98.77796	0.98785
ClimateModel	Borderline SMOTE	89.03592	92.89941	90.92664	90.85564	0.90874
ClimateModel	MWMOTE	98.34711	76.28205	85.92058	82.99566	0.84194
ClimateModel	SCOT	**98.8024**	**100**	**99.39759**	**99.39939**	**0.99401**
OCR	Raw	4.52128	**100**	8.6514	21.26329	0.52261
OCR	SMOTE	98.69452	**100**	99.34297	99.34511	0.99347
OCR	Borderline SMOTE	98.25936	**100**	99.12204	99.12586	0.9913
OCR	MWMOTE	**99.39007**	**100**	**99.6941**	**99.69457**	**0.99695**
OCR	SCOT	96.88202	**100**	98.41632	98.42867	0.98441
OilSpill	Raw	0	NaN	NaN	0	0.5
OilSpill	SMOTE	97.54464	**100**	98.75706	98.76469	0.98772
OilSpill	Borderline SMOTE	93.81003	97.34219	95.54348	95.5496	0.95566
OilSpill	MWMOTE	97.76286	92.38901	95	94.81926	0.94864

Table B-3

Cont.

OilSpill	SCOT	**99.77728**	**100**	**99.88852**	**99.88858**	**0.99889**
Ozone	Raw	0	NaN	NaN	0	0.5
Ozone	SMOTE	98.49974	**100**	99.2442	99.24704	0.9925
Ozone	Borderline SMOTE	98.47894	98.55582	98.51736	98.20615	0.98207
Ozone	MWMOTE	98.56968	93.09147	95.75228	94.19035	0.94288
Ozone	SCOT	**99.83786**	**100**	**99.91886**	**99.9189**	**0.99919**
PageBlocks	Raw	81.38528	23.5	36.46945	84.7842	0.84855
PageBlocks	SMOTE	97.44372	95.28073	96.35009	96.30197	0.96309
PageBlocks	Borderline SMOTE	96.60924	93.37467	94.96442	94.71878	0.94737
PageBlocks	MWMOTE	98.33779	97.42571	97.87962	97.87057	0.97872
PageBlocks	SCOT	**99.3646**	**97.84689**	**98.59991**	**98.56308**	**0.98566**
Statlog	Raw	78.11502	68.39161	72.93065	86.64637	0.87112
Statlog	SMOTE	**99.94836**	97.80997	**98.8676**	**98.84918**	**0.98855**
Statlog	Borderline SMOTE	97.94206	96.05651	96.99012	96.77785	0.96785
Statlog	MWMOTE	99.19105	98.11032	98.64772	98.63857	0.9864
Statlog	SCOT	98.7766	**98.11886**	98.44663	98.41501	0.98416
RobotNav	Raw	65.85366	96.86099	78.4029	81.09486	0.82859
RobotNav	SMOTE	**99.922**	99.82466	**99.87331**	**99.87323**	**0.99873**
RobotNav	Borderline SMOTE	98.54646	99.79504	99.16682	99.16404	0.99166
RobotNav	MWMOTE	99.74604	**99.88263**	99.81429	99.8145	0.99815
RobotNav	SCOT	99.21214	99.8646	99.5373	99.53729	0.99538
Mammography	Raw	83.07692	32	46.20321	89.21094	0.89437
Mammography	SMOTE	93.72883	96.03227	94.86657	94.92056	0.94928
Mammography	Borderline SMOTE	99.00653	97.31679	98.15439	98.10301	0.98107
Mammography	MWMOTE	99.39566	97.92512	98.65491	98.64215	0.98645
Mammography	SCOT	**99.80051**	**97.97045**	**98.87701**	**98.85208**	**0.98857**

Table B-3

Cont.

Seismic	Raw	15.88235	11.68831	13.46633	38.13159	0.53716
Seismic	SMOTE	90.76222	92.95715	91.84657	91.93525	0.91943
Seismic	Borderline SMOTE	93.81679	78.70637	85.59986	82.44543	0.83135
Seismic	MWMOTE	**99.25589**	56.03267	71.62888	46.68178	0.60606
Seismic	SCOT	97.96251	**98.48423**	**98.22268**	**98.21457**	**0.98215**
Covertype	Raw	97.16054	78.36171	86.75443	97.54884	0.9755
Covertype	SMOTE	99.84617	98.52892	99.18317	99.17546	0.99178
Covertype	Borderline SMOTE	99.45722	98.70921	99.0818	99.03408	0.99035
Covertype	MWMOTE	**99.88815**	98.52982	99.20433	99.19631	0.99199
Covertype	SCOT	99.80079	**98.79756**	**99.29664**	**99.28514**	**0.99286**

Table B-4

Wilcoxon Signed-Rank test to compare the geometric mean values when applying SMOTE, Borderline SMOTE, MWMOTE, and SCOT to twelve real world datasets for the decision tree simulation. The best results for each dataset are highlighted in bold.

Data set	Original Data	SCOT vs. SMOTE			SCOT vs. B-SMOTE			SCOT vs. MWMOTE		
		SCOT	SMOTE	Rank	SCOT	BSMOTE	Rank	SCOT	MWMOTE	Rank
Abalone	60.49991	**98.55427**	90.92862	11.0	**98.55427**	89.02128	11.0	**98.55427**	96.50710	10.0
ClimateModel	53.63893	**97.99109**	92.09436	10.0	**97.99109**	84.27297	12.0	**97.99109**	95.30129	11.0
CoverType	91.91633	99.08596	98.90634	3.0	99.08596	98.80689	2.0	99.08596	**99.16771**	-4.0
Mammography	73.89331	**99.23431**	96.70047	6.0	**99.23431**	98.17074	4.0	**99.23431**	99.08439	6.0
OCR	97.0497	99.50194	99.33275	2.0	99.50194	99.36173	1.0	99.50194	**99.60797**	-5.0
OillSpill	51.41984	**99.72134**	96.53702	7.0	**99.72134**	94.96390	9.0	**99.72134**	97.98882	8.0
Ozone	50.73731	**99.39077**	96.06386	8.0	**99.39077**	94.96919	8.0	**99.39077**	97.59617	9.0
Page Blocks	85.12482	**99.12084**	98.41632	4.0	**99.12084**	96.73619	6.0	**99.12084**	99.04542	2.0
Robot Nav.	97.79487	**99.94192**	99.89275	1.0	**99.94192**	99.62264	3.0	**99.94192**	99.86337	3.0
Seismic	37.36665	**96.59236**	92.54349	9.0	**96.59236**	93.87889	7.0	**96.59236**	94.97208	7.0
Statlog	71.3394	95.22935	94.23872	5.0	95.22935	93.74252	5.0	95.22935	**95.25462**	-1.0
TCC	27.15749	**98.78978**	84.25890	12.0	**98.78978**	89.34189	10.0	**98.78978**	94.78594	12.0
		T = min{78, 0} = 0			T = min{78, 0} = 0			T = min{68, 10} = 10		

Table B-5

Wilcoxon Signed-Rank test to compare the geometric mean values when applying SMOTE, Borderline SMOTE, MWMOTE, and SCOT to twelve real world datasets for the support vector machine simulation. The best results for each dataset are highlighted in bold.

Data set	Original Data	SCOT vs. SMOTE			SCOT vs. B-SMOTE			SCOT vs. MWMOTE		
		SCOT	SMOTE	Rank	SCOT	BSMOTE	Rank	SCOT	MWMOTE	Rank
Abalone	61.84998	**99.63901**	94.72290	10.0	**99.63901**	91.74003	10.0	**99.63901**	88.53030	10.0
ClimateModel	0	**99.39939**	98.77796	4.0	**99.39939**	90.85564	11.0	**99.39939**	82.99566	11.0
CoverType	97.54884	**99.28514**	99.17546	1.0	**99.28514**	99.03408	1.0	**99.28514**	99.19631	1.0
Mammography	89.21094	**98.85208**	94.92056	9.0	**98.85208**	98.10301	4.0	**98.85208**	98.64215	2.0
OCR	21.26329	98.42867	99.34511	-6.0	98.42867	99.12586	-3.0	98.42867	**99.69457**	-6.0
OillSpill	0	**99.88858**	98.76469	7.0	**99.88858**	95.54960	8.0	**99.88858**	94.81926	8.0
Ozone	0	**99.91890**	99.24704	5.0	**99.91890**	98.20615	6.0	**99.91890**	94.19035	9.0
Page Blocks	84.7842	**98.56308**	96.30197	8.0	**98.56308**	94.71878	7.0	**98.56308**	97.87057	5.0
Robot Nav.	81.09486	99.53729	**99.87323**	-2.0	99.53729	99.16404	2.0	99.53729	99.81450	-4.0
Seismic	38.13159	**98.21457**	91.93525	11.0	**98.21457**	82.44543	12.0	**98.21457**	46.68178	12.0
Statlog	86.64637	98.41501	**98.84918**	-3.0	98.41501	96.77785	5.0	98.41501	98.63857	-3.0
TCC	36.59136	**99.75816**	89.02929	12.0	**99.75816**	92.49946	9.0	**99.75816**	96.54391	7.0
		T = min{67, 11} = 11			T = min{75, 3} = 3			T = min{65, 13} = 13		