5-2020

Knot Flow Classification and its Applications in Vehicular Ad-Hoc Networks (VANET)

David Schmidt
East Tennessee State University

Knot Flow Classification and its Applications in Vehicular Ad-Hoc Networks (VANET)

A thesis
presented to
the faculty of the Department of Computing
East Tennessee State University

In partial fulfillment
of the requirements for the degree
Master of Science in Computer and Information Sciences, Applied Computer Science

---

by
David Allan Schmidt
May 2020

Mohmmad S. Khan, Brian T. Bennett
Matthew Harrison
Edward Hall

Keywords: Machine Learning, Internet of Things, Internet of Vehicles, Intrusion Detection, Knot Flow Classification, Data Science, Vehicular Ad-hoc Networks, Wireless Computing

ABSTRACT

Knot Flow Classification and its Applications in Vehicular Ad-Hoc Networks (VANET)

by

David Allan Schmidt

Intrusion detection systems (IDSs) play a crucial role in the identification and mitigation for attacks on host systems. Of these systems, vehicular ad hoc networks (VANETs) are difficult to protect due to the dynamic nature of their clients and their necessity for constant interaction with their respective cyber-physical systems. Currently, there is a need for a VANET-specific IDS that meets this criterion. To this end, a spline-based intrusion detection system has been pioneered as a solution. By combining clustering with spline-based general linear model classification, this knot flow classification method (KFC) allows for robust intrusion detection to occur. Due its design and the manner it is constructed, KFC holds great potential for implementation across a distributed system. The purpose of this thesis was to explain and extrapolate the afore mentioned IDS, highlight its effectiveness, and discuss the conceptual design of the distributed system for use in future research.

DEDICATION

This thesis is dedicated to my wonderful significant other, Taylor Miller, of whom if I were without none of this would be possible. I would also like to dedicate this thesis to my friends and faculty at East Tennessee State University's Department of Computing for inspiring me to push forward.

ACKNOWLEDGEMENTS

I would first like to thank my thesis advisor Dr. Mohammad Khan and co-advisor Dr. Brian Bennett for their support during the creation of this thesis. Both challenged me to push my understanding of mathematics, statistics, machine-learning, and programming to develop something truly unique. I sincerely appreciate the time they have both given to aid me in my work, as well as the guidance they provided during times of uncertainty.

I would also like to thank the Jean-Marie Nshimiyimana for his assistance in the early stages of this thesis. Without Jean's support, the early task of visualizing the data for this experimentation would have been much more difficult.

An acknowledgement to Matthew Harrison and Edward Hall for their support as part of my thesis committee where both their editing and mindful considerations have helped me put these concepts forward.

Finally, a special thanks to Tomas Hill and Austin Helton for assisting in the optimization of what would become the Knot Flow Classification algorithm.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1. INTRODUCTION

1.1 Motivation

As the world becomes increasingly interconnected through the expansion of the internet, the movement from wired to wireless networks has become prominent as the next step in the evolution of human communication. With this wireless communication, the implementation of networking has evolved as well. Traditional networks, consisting of static machines and their respective servers, have given way to the development of modern networks comprising of mobile devices, appliances, and other nontraditional mediums [1] [2]. Of these mediums, autonomous vehicles and the study of their interlocking wireless systems have become the focus for research in artificial intelligence, machine-learning, and traffic-control networking [3] [4] [5] [6]. Specifically, the spontaneous creation of networks of these ad-hoc networks provide an interesting challenge when applied to a vehicular domain. These vehicular ad-hoc networks (VANETs) are a testament to the way in which travel, commerce, and business are to be conducted in the coming future [3] [7].

Although the means in which communication occurs has evolved, the use of an intrusion detection system (IDS) that classifies, validates, and verifies the integrity of the data within a VANET is still needed just as in static client-server network environments [8]. However, the VANET environment in which an IDS must operate has proved challenging due to the sheer volume of data that must be processed and the high speed in which this data must be completely and correctly verified [4] [9]. To be both effective and efficient, a VANET-focused IDS must handle these large amounts of data and provide expedient results, otherwise the damaging of property, injury of human beings, and loss of human life may occur.

1.2 Statement of Research Problem

Initially, this research focused on testing the feasibility of a novel VANET specific IDS that utilizes basis splines for the classification of malicious network activity from normal network communications [10]. The research was subsequently expanded into the development of a VANET specific machine-learning algorithm. The success of this application, as well as the need for more subsequent testing of the IDS with larger and more complex sets of data, spurred research forward. Efforts to develop a VANET specific IDS for use in autonomous vehicles have been ongoing since late 2010. Unfortunately, current implementations of these systems are either inefficient or inaccurate for use in safety-critical systems [8]. These limitations have created a need for an IDS that is both effective and efficient within a VANET environment and for autonomous vehicle applications.

Accordingly, the work described in this document focused on three related challenges and their possible solutions:

1. *Developing, testing, and analyzing the use of basis splines to classify network activity specific to that of a VANET environment.* This application uses basis spline variants to classify malicious network activity. The system was tested using 120 observations from a VANET environment developed by E. A. Shams, A. Rizaner and A. H. Ulusoy from the Eastern Mediterranean University in Turkey [11] (see chapter 2).
2. *Developing, testing, and analyzing the accuracy and speed of an ensemble classification method that utilizes basis splines and dynamic clustering for intrusion detection.* This challenge led to the development of a machine-learning algorithm, termed knot flow classification (KFC), that dynamically clusters data, contracts the clusters towards their respective centers, and subsequently classifies data via the basis spline IDS application.

The proposed algorithm uses over 22,544 observations from the KDD-NSL dataset created by the University of New Brunswick in Canada [12] [13] (see chapter 3).

3. *Partitioning the aforementioned classification method across a distributed system and analyzing its effects to the accuracy and speed of classification.* The conceptualization of Knot Flow Classification across a distributed system using Map Reduce and Hadoop [14]. The proposed architecture for this system is discussed at the end of this thesis (see chapter 4).

1.3 Results

The research described in this thesis produced three novel results.

1. Confirmation that basis splines are effective when classifying malicious network activity from normal VANET network data. Additionally, the comparison of data of the dataset utilized in this experimentation mirrored well to the NSL-KDD dataset, allowing for more extensive future testing.
2. Confirmation that KFC was applicable, effective, and efficient when processing network traffic data from a simulated VANET environment.
3. Confirmation that KFC was as accurate as modern classification techniques with the addition of being more efficient when processing datasets larger than 13,000 observations.

These results, together with other considerations, such as avenues for future research and a detailed description of the KFC distributed system architecture, are described in more detail in this thesis's remaining chapters.

CHAPTER 2. SPLINE BASED INTRUSION DETECTION IN VEHCICULAR AD HOC NETWORKS (VANET)

David A. Schmidt, Mohammad S. Khan, Brian T. Bennett

2.1 Abstract

Intrusion detection systems (IDSs) play a crucial role in the identification and mitigation for attacks on host systems. Of these systems, vehicular ad hoc networks (VANETs) are particularly difficult to protect due to the dynamic nature of their clients and their necessity for constant interaction with their respective cyber-physical systems. Currently, there is a need for a VANET-specific IDS that can satisfy these requirements. Spline function-based IDSs have shown to be effective in traditional network settings. By examining the various construction of splines and testing their robustness, the viability for a spline-based IDS can be determined.

Keywords- Intrusion Detection, Spline, Vehicular Ad Hoc Networks, Machine Learning, Internet of Things, Internet of Vehicles

2.2 Introduction

The internet of things (IoT) has become an area of interest as society begins to connect a vast number of computer systems to the internet. Due to this societal switch and consequent volume of information becoming digitalized, intrusion detection has become a high-priority concern. With the implementation of so many types of computer systems, attempts have been made to create an optimal intrusion detection system (IDS) for each domain in IoT. Among these domains, the internet of vehicles (IoV), consisting of vehicular ad hoc networks (VANETs), has proven to be an especially difficult domain due to the dynamic nature of their clients and the complexity of the criteria associated with its optimal IDS. Currently, the use of spline functions

within IDSs has been pioneered in multiple domains with various levels of success. The purpose of this article is to examine the role of an IDS, the obstacles associated with VANET, and define a preliminary spline-based IDS configuration for a VANET.

2.3 Purpose of an Intrusion Detection System

An Intrusion Detection System (IDS) is an application that identifies attacks against a host system. IDSs identify these attacks by processing system-centric events using various machine learning and data mining techniques. Ideally, these techniques allow an IDS to function efficiently and effectively.

The effectiveness of an IDS stems from its ability to distinguish between normal processing and attacks, the speed with which it identifies attacks, and how well it determines an attack's type. At minimum, an IDS should distinguish between denial of service attacks, probes, unauthorized elevation of privilege, and remote access attacks. Efficiency results from lowering the time required for processing and the resources required for these identifications to occur. Overall, an optimal IDS should balance efficiency and effectiveness to remain robust.

2.3.1 Probing

Sharma and Kaul state “The most common type of attack, a probing is an attack that monitors a target system to collect data and identify weaknesses” [15]. If an attacker can exploit a weakness, they can use this information to compromise the integrity of a system. A myriad of probing attacks exists. Often, these attacks involve the exploitation of a system’s hardware, such as an open-access port. More often, probing attacks are the result of a user’s incompetence and sensitive information is divulged to the attacker.

2.3.2 Denial of Service (DoS) Attacks

Denial of Service (DoS) is a type of attack where an attacker overburdens system resources [15]. This prevents users from making legitimate requests from one of the compromised resources, thus, denying access. DoS typically occurs when an attacker abuses a feature of a system by exploiting bugs or poor designs within the system. “Often, these types of attacks are classified based on the resource that is compromised” [15]. For instance, a UPD socket flood is a DoS attack that floods a target with packets. This gridlocks a system because it is unable to process every request.

2.3.3 User to Root (U2R) Attacks

User to Root (U2R) attacks are a type of attack where an intruder attempts to gain root access to a target system” [15]. Once an attacker gains root access, they can obtain administrator privileges, thus compromising the system and the integrity of its contents. This type of attack usually occurs in conjunction with a buffer-overflow exploit but can also be found in attacks such as code injection techniques.

2.3.4 Remote to User (R2U) Attacks

“Remote to User (R2U) attacks are a type of attack where an attacker exploits a system over a network by sending malicious packets in order to expose the target system to vulnerabilities” [15]. The attacker then exploits the target system to gain user access and exploit vulnerabilities as a local user. Typically, this type of attack occurs as phishing; however, this attack may also occur if an attack alters networking control protocols.

2.4 Obstacles Associated with Vehicular Ad-hoc Networks

Unlike traditional networks which consist of simple static client-server relationships, VANET’s relationships have an increased complexity. This is due to the dynamic nature of their

clients, the multifaceted nature of their cyber-physical systems, and the safety-critical nature of the environments in which they are implemented. Within the system, fast-moving autonomous vehicles act as clients who, in turn, interact with a cyber-physical system consisting of cellular service towers, roadside units, and other clients within range. Development of an optimal IDS will not only need to account for the client's safety in terms of the five types of attacks, but also the integrity of the cyber-physical system. Outside the realm of cyber-attacks, damage to power-grid infrastructure and roadside units via physical attacks will need to be accounted for as well for a VANET IDS [16]. Any damage received by the cyber-physical system may lead to inconsistencies in communication. If these inconsistencies are not corrected, the resulting communication breakdown may result in a breach of security.

Due to the need for constant communication between a vehicle and the cyber-physical system, network latency is a high priority concern. There are many components that must be accounted for to provide a fast, responsive connection. However, the limiting factor for an optimal connection is in essence how fast a client can detect an attack and the speed in which it communicates to the corresponding system to mitigate any damage [17]. In a situation where a vehicle is traveling at high speeds or even at lower speeds in densely populated areas, the health and safety of individuals are at stake. This safety-criticality must not be taken lightly. Calculated decisions must be made within fractions of a second, always with preservation of life as a top priority. Failure to meet this criterion is a failure to meet the standards of an optimal IDS.

2.4.1 Splines

A spline is a mathematical depiction of a continuous function consisting of points, called knots, that allows the user to manipulate the shape of curves [18]. A user interfaces with a spline function by entering a specific number of knots. A curve is then created between each of these

knots. A curve that passes through one of these knots is deemed an interpolating curve and one that passes near them is deemed an approximation curve [19]. Due this ability, spline functions are useful when shaping two-dimensional and, in the case of B-Splines, three-dimensional depictions of functions up to the complexity of a cubic polynomial [20]. Any computation in a higher complexity environment can result in a loss of accuracy due to a decrease in the amount of interpolation corresponding with an increase in approximation [21]. Based on their continuity, splines can be organized into different subsets including, but not limited to, linear, quadratic, cubic, and basis splines.

2.4.2 Linear Splines

A linear spline is the simplest form of interpolation. It is constructed piece-wise from linear functions creating two–point interpolating polynomials.

An example of a linear interpolating spline (Fig. 1) and an equation (1) representing a linear spline interpolation, as seen in [22], can be seen below.

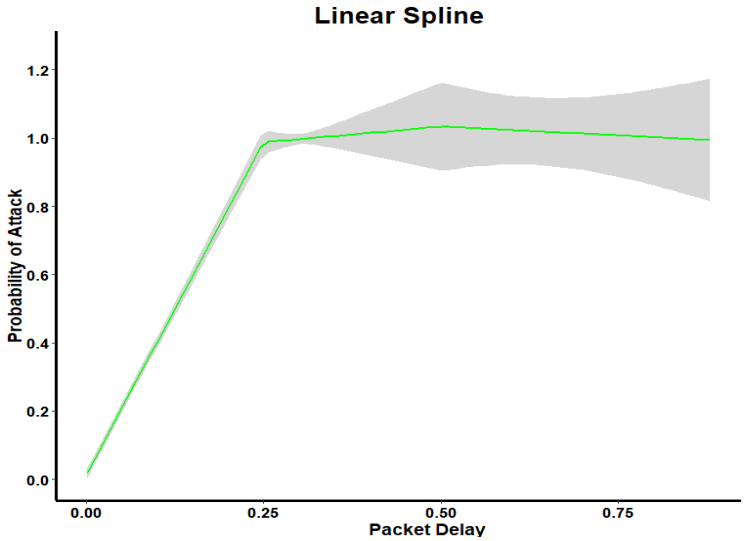

Fig. 1 Visual representation of a linear spline. Grey denotes three degrees of freedom. Knot placement occurs at the 0.25, 0.50, and 0.75 packet delay quantile.

A linear spline g is given by

$$g(x) = \sum_{i=i}^{m} y_i B_{i,i}(x)$$

And satisfies the interpolation conditions

$$g(x_i) = y_i, \text{ for } i = 1, \ldots, m-1 \text{ and } \lim_{x \to x_m^-} g(x) = y_m \quad (1)\ [22]$$

2.4.3 Linear Spline Interpretation

Due to its two-point polynomial interpolation, linear splines are confined to a strict degree of freedom when interpolating data. These degrees of freedom are derived from the variable placement of knots during the spline's construction which, in turn, are confined by its interpolation conditions [18]. As demonstrated in Fig. 1, these conditions allow polynomials only of the second degree to be utilized, thus restricting the possible spline variations. This variation, although minimal, is key to the spline's robustness when classifying data as it allows for placement of the best fit linear spline.

2.4.4 Quadratic Splines

Increasing in complexity, quadratic spline construction is like that of linear splines. However, rather than consisting of piece-wise linear functions, it is constructed from piece-wise quadratic functions. An example of a quadratic spline (Fig. 2) and an equation (2) representing its piece-wise construction, as demonstrated in [23], can be seen below.

The quadratic spline $S_{2,2}(x)$ is constructed as:

$$P_1(x) = a_1 + b_1 x + c_1 x^2, \quad on \quad [-1, 0]$$
$$P_2(x) = a_1 + b_1 x + c_1 x^2, \quad on \quad [0, 1]$$

And $S_{2,2}(x)$ interpolates the given data points,

$$P_1(-1) = a_1 + b_1 + c_1 = 0$$
$$P_1(0) = a_1 = 1$$
$$P_1(0) = a_2 = 1$$
$$P_1(1) = a_2 + b_2 + c_2 = 3$$

The quadratic spline function is given as

$$S_{2,2}(x) = \begin{cases} 1 + 2x + x^2, & on\ [-1,0] \\ 1 + 2x, & on\ [-1,0] \end{cases} \qquad (2)\ [23]$$

2.4.5 Quadratic Spline Interpretation

Like linear splines, quadratic splines also possess a varying degree of freedom. However, due to the nature of their construction, quadratic splines can interpolate knots at a second degree which increases the possible interpolant variations for the spline [18]. As seen in Fig. 2, the quadratic spline contains a much wider area of coverage than the linear spline with identical degrees of freedom. Based on the placement of its knots, the increased freedom of the quadratic spline can specify the way it interpolates knots, thus increasing its robustness when classifying data.

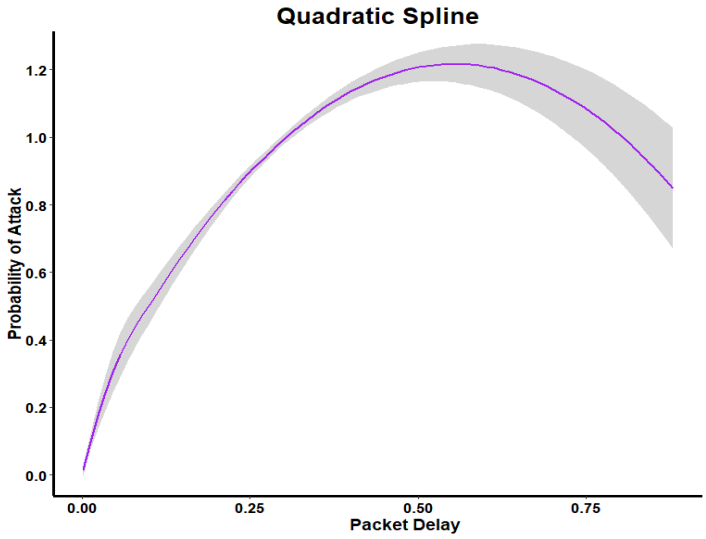

Fig. 2 Visual representation of a quadratic spline. Grey shading denotes three degrees of freedom. Knot placement occurs at the 0.25, 0.50, and 0.75 packet delay quantile.

2.4.6 Cubic Splines

The highest level of complexity cubic spline construction is like that of linear splines, but rather than consisting of piece-wise linear functions, it is constructed from piece-wise cubic functions. An example of a cubic spline (Fig. 3) and an equation (3) representing a cubic spline's construction, as seen in [23], can be seen below.

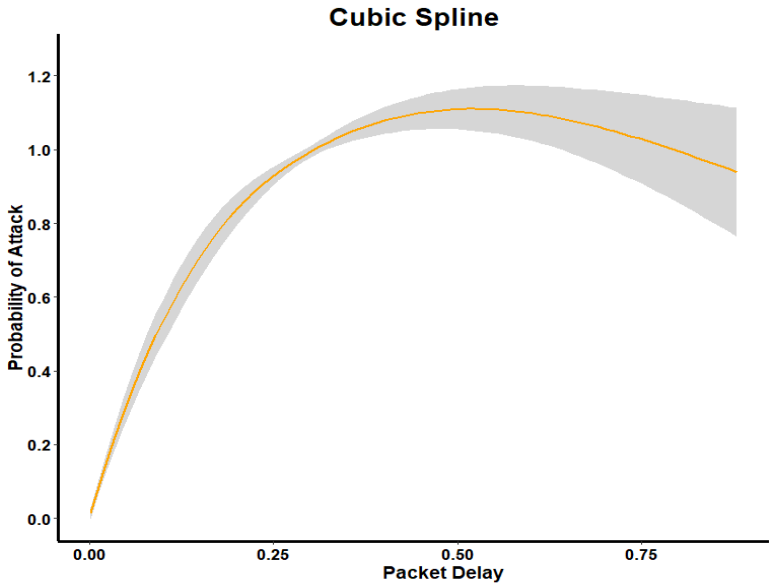

Fig. 3 Visual representation of a cubic spline. In this instance. Grey shading denotes three degrees of freedom. Knot placement occurs at the 0.25, 0.50, and 0.75 packet delay quantile

Assuming the data used for construction is $\{(x_i, f_i)\}_{i=0}^{n}$, and $S_{3,n}(x)$ is a cubic spline constructed as:

$$\begin{aligned} P_1(x) &= a_1 + b_1 x + c_1 x^2 + d_1 x^3, \qquad x \in [x_0, x_1], \\ P_2(x) &= a_2 + b_2 x + c_2 x^2 + d_2 x^3, \qquad x \in [x_1, x_2], \\ &\vdots \\ P_n(x) &= a_n + b_n x + c_n x^2 + d_n x^3, x \in [x_{n-1}, x_n]; \end{aligned}$$

which interpolates the given data points

$$S_{3,n}(x) = f_i, \qquad i = 0, 1, \ldots, n$$

the cubic spline function is given as

$$S_{3,n}(x) = \begin{cases} P_1(x) = a_1 + b_1 x + c_1 x^2 + d_1 x^3, & x \in [x_0, x_1] \\ P_2(x) = a_2 + b_2 x + c_2 x^2 + d_2 x^3, & x \in [x_1, x_2], \\ \quad . & \\ \quad . & \\ P_n(x) = a_n + b_n x + c_n x^2 + d_n x^3, & x \in [x_{n-1}, x_n] \end{cases} \quad (3)\ [23]$$

2.4.7 Cubic Spline Interpretation

In comparison to the previously mentioned splines, cubic splines demonstrate the highest level of variation within the same degree of freedom. This is due to the spline's piecewise cubic construction which allows interpolation between knots to occur at the third degree, further increasing the number of possible variations for the spline [18]. With this expanded variation, cubic splines can further manipulate the manner in which they interpolate knots and further increasing its robustness when classifying data.

2.4.8 Basis Splines (B-Splines)

A B-spline is a special case of spline function in which a spline of order n in a piece-wise constructed function of the degree *n – 1* in terms of a variable *x* [19]. If a B-spline of this order is equivalent among all knots, all possible spline functions for the set of polynomials pertaining to the B-spline can be constructed using a combination of linear B-splines with only a single unique combination for each spline [19]. An equation (4) depicting a mathematical representation of a B-spline and its construction, as demonstrated in [24], can be seen below.

A B-spline curve *P(t)* is defined by

$$P(t) = \sum_{i=0}^{n} P_i N_{i,k}(t)$$

Where:

- the control points are $\{P_i : i = 0, 1, \ldots, n\}$,
- k is the order of polynomial segments within the curve,
- the $N_{i,k}(t)$ are the normalized B-spline blending functions described by the order k and by a non-decreasing order of real numbers

$$\{t_i : i = 0, \ldots, n + k\}.$$

The $N_{i,k}$ component functions are

$$N\_(i, 1)\ (t)\begin{cases} 1 \text{ if } u \in [t_i, t_{i+1}), \\ 0 \quad \text{otherwise} \end{cases}$$

Where, if $k > 1$,

$$N_{i,k}(t) = \frac{t - t_i}{t_{i+k-i} - ti} N_{i,k-1}(t) + \frac{t_{i+k} - t}{t_{i+k} - t_{i+1}} N_{i+1,k-1}(t) \text{ (4) [23].}$$

2.5 Concept for VANET Splined-Based IDS

Although the concept of a spline-based IDSs has been implemented using traditional client-to-sever relationships [20] [21], there are limited cases of its use in IoT and few, if any, utilized within the realm of IoV. Recently, experimentation within the domain of IoV conducted by Shams, Rizaner, and Ulusoy [11] has provided a novel approach for intrusion detection using a support vector machine (SVM) in combination with a trust value table (TVT). Due to the success of this approach, the implementation of splines to this framework may further optimize the IDS. This optimization stems from the nature of the SVM where a hyper-plane is created at an optimal distance between adjacent data points [25]. It is speculated that the addition of splines may further optimize this process, increasing the accuracy when identifying malicious attacks.

2.6 Experimentation

In order to demonstrate the viability of a spline-based IDS in an IoV environment, several spline regressions, as well as a logistic regression, were implemented on an IoV dataset provided

by Shams et al. [11] (Fig 4). The data consisted of network traffic between fifty-two simulated autonomous vehicles that monitored packet delay, the number of packets dropped, and the frequency of transfer interval in both congested and non-congested environment [15]. Of these data, packet delay was selected as the independent variable due to its ability to perform as an attack predictor. Six hundred observations were randomly selected and split into subsets with an 80:20 ratio for training and testing. Confusion-matrix analysis for the splines demonstrated accuracies greater than 94%, with the B-spline holding an accuracy of 95.83% like that of the 96.67% accuracy of the logistic regression (Table 1).

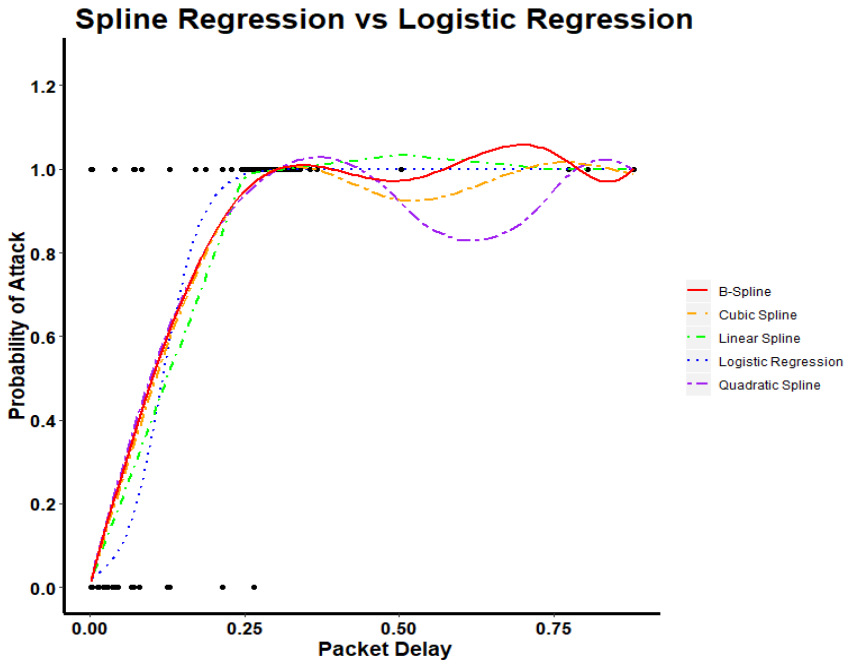

Fig 4 Prediction of attack, indiscriminate of type, using various splines and logistic regression. Knot placement occurs at the 0.25, 0.50, and 0.75 packet delay quantile

Table 1 Confusion matrix analysis for logistic and spline regressions, where N denotes the number of observations.

N = 120	True Positive	False Positive	True Negative	False Negative	Prediction Accuracy
Logistic Regression	58	2	58	2	96.67%
Linear Spline	57	1	58	4	95.83%
Quadratic Spline	57	3	56	4	94.17%
Cubic Spline	57	1	58	4	95.83%
B-Spline	56	3	57	4	95.83%

2.7 Conclusion

IDSs designed for vehicular ad hoc networks must be able to compensate for the dynamic nature of their clients, their associated safety criticality, and the multifaceted nature of their cyber-physical systems in order to be effective. Due to their ability to interpolate curves, their high levels of efficiency, and their success in traditional network environments, spline functions may prove to be vital component for an optimal VANET IDS. The use of B-splines and their ability to form unique combinations of linear functions are of extreme interest as they may provide a robust solution to mediate these obstacles. In conjunction with the ensemble of SVM and TVT, other techniques such as association and clustering analysis may also provide valuable insight for the construction of an optimal IDS, as they may allow the grouping of specific types of attacks, leading to faster classification and, therefore, more robust decision making.

2.8 References

[15] S. Sharma and K. A., "A survey on Intrusion Detection Systems and Honeypot based proactive security mechanisms in VANETs and VANET cloud," *Vehicular Communications,* vol. 12, pp. 138-164, 2018.

[16] A. Ashok, M. Sovindarasu and J. Wang, "Cyber-physical attack-resilient wide-area monitoing, protection, and control of the power grid," *IEEE Proceedings,* 2017.

[17] F. Sakiz and S. Sen, "A survey of attacks and detection mechanisms on intelligent transportation systems: VANETs and IoV," *Ad Hoc Networks,* vol. 61, pp. 33-50, 2017.

[18] L. L. Shumaker, Spline Functions : basic theory, Cambridge: Cambridge Printing House, 1981.

[19] "Spline Curves," Clemson University, [Online]. Available: https://people.cs.clemson.edu/~dhouse/courses. [Accessed 10 10 2018].

[20] S. Mukkamala, A. Sung, A. Ajith and V. Ramos, "Intrusion Detection Using Adaptive Regression Splines," *Enterprise Information Systems,* vol. 6, pp. 211-218, 2006.

[21] H. B. Lu and Q. L. Wang, "Intrusion detection based on spline neural network," *Jisuanji Yingyong Yanjiu Application Research of Computers,* vol. 26, no. 2, pp. 448-450, 2009.

[22] "Spline Approximation of Fuctions and Data," University of Oslo, [Online]. Available: https://www.uio.no/studier/emner/matnat/ifi/nedlagte-emner/INF-MAT5340/v05/undervisningsmateriale/kap5-new.pdf. . [Accessed 28 12 2018].

[23] U. o. Houston, "Interpolation By Splines," [Online]. Available: https://www.math.uh.edu/~jingqiu/math4364/spline.pdf . [Accessed 28 10 2018].

[24] K. I. Joy, "Definition of B-Spline Curve," University of California Davis, [Online]. Available: https://www.cs.unc.edu/~dm/UNC/COMP258. [Accessed 1 1 2018].

[25] G. James, T. Hastie and R. Tibshirani, An Introduction to Statistical Learning with Applications in R, New York: Springer, 2013.

CHAPTER 3. SPLINE-BASED INTRUSION DETECTION FOR VANET UTILIZING KNOT FLOW CLASSIFICATION

David A. Schmidt, Mohammad S. Khan, Brian T. Bennett

3.1 Abstract

Intrusion detection systems (IDSs) are an integral component for the identification and mitigation of attacks on computing systems. Of these systems, vehicular ad hoc networks (VANETs) are particularly difficult to protect due to the dynamic nature of their clients and the volume of information passed between them and their respective infrastructure. To meet these requirements, a spline-based intrusion detection system has been pioneered as a prospective solution. By combining clustering with spline based general linear model classification, this knot flow classification method (KFC) allows robust intrusion detection to occur.

Keywords- Internet of Things, Vehicular Ad Hoc Networks, Intrusion Detection, Internet of Vehicles, Clustering, Splines, Machine Learning, NSL-KDD, Knot Flow Classification

3.2 Introduction

With an ever-increasing number of computer systems and devices becoming interconnected, the internet of things automated network system has become an area of study for a variety of domains [2]. Often, the data associated within these domains is sensitive in nature and must be protected through the use of an intrusion detection system (IDS) [8]. Intrusion detection systems have been implemented using a variety of statistical, machine-learning, or other artificial intelligence techniques to identify, contain, and prevent potential malicious events [9] [15]. However, these techniques can be limited by the dynamic nature of their domains causing complications during computation which diminish efficiency and effectiveness. Of these domains, vehicular ad hoc networks (VANETs) offer a unique environment where computation

time, accuracy, and reliability of the IDS are paramount to human safety. Previous studies have noted that spline based intrusion detection for VANET environments are viable, however, highly complex splines may diminish the efficiency of an IDS, thus reducing its effectiveness [10]. The purpose of this article is to address this concern and demonstrate the use of knot flow classification (KFC) as a means to mediate this pitfall.

3.1 Ensemble Classification

Knot flow classification is conducted as a form of ensemble classification, where the result of dynamic clustering provides the basis for spline implementation [26]. The reasoning behind this process is two-fold. Data originating from real-time environments is often dense and unruly to work with. When a spline is implemented in this type of environment, it is forced to make a variety of complex curves in order to perform accurate classification. As the complexity of a spline increases (e.g. the spline increases in dimensionality), the computation time needed for classification grows in tandem [27]. By first segregating the data into clusters, these curvatures can be reduced thus decreasing computation time.

3.2 Spline Utilization and Computation Time

Spline implementation utilizes the placement of interconnected points providing a means of classification. These points, also known as knots, are placed either statically or dynamically to best fit the data [18].With either types of placement, there are both advantages and disadvantages. Static placement of knots may result increased robustness during classification, keeping the computation time low at the cost of accuracy [27]. Dynamic placement of knots maintains a high level of robustness. However, the nature of dynamic knot placement may lead to an overabundance of knots being placed, increasing computation time. Dynamic knot

placement may also create what is known as a "false" knot which may lead to a substantial loss in classification accuracy [18] [28].

3.3 Dynamic Clustering

Dynamic clustering within KFC utilizes the k-means clustering algorithm to perform vector quantization. K-means partitions data into a number of specified clusters in which each data point belongs to the cluster with the nearest mean [13]. The number of clusters is predetermined by using the "Elbow" method allowing for the optimal number of clusters to be selected [28]. The standard algorithm in which k-means clustering occurs consists of two altering steps. The assignment step where, given an initial set of k-means clusters c_i each observation is assigned to a cluster who means is the least squared Euclidean distance:

$$S_i^{(t)} = \{x_p : ||x_p - c_i^{(t)}||^2 \leq ||x_p - c_j^{(t)}||^2 , 1 \leq j \leq k, 1 \leq i \leq j$$

where every observation x_p is assigned to a single cluster $S^{(t)}$. The update step then calculates the new mean distance from the new mean distance of the observations to their respective cluster centers:

$$c_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j$$

until observation assignment no longer alters the means of any cluster [13].

3.4 Cluster Contraction

One of the novel facets of KFC is the ability to contract the points of each cluster towards their designated cluster centers:

$$P_{(x,y)} \in C = P_{(center_x + a*(x_i - center_x)),(center_y + a*(y_i - center_y))}$$

Where $center_x$ is the x coordinate for the cluster center, $center_y$ is the y coordinate for the cluster center, a is the percentage of reduction, x_i and y_i represent each point of that cluster. This contraction can also be performed in a three-dimensional space:

$$P_{(x,y,z)} \in C = P_{(center_x + a*(x_i - center_x),(center_y + a*(y_i - center_y),(center_z + a*(z_i - center_z))}$$

contraction in this manner results in the formation of a perimeter of empty space between each cluster. When used in tandem with splines, this perimeter provides a pathway that can be utilized by the spline, reducing the need for complex twists and turns. With this reduction in complexity, knot placement for a spline can be reduced. Moreover, this "knot horizon" removes the need for dynamic knot selection and static knot selection can be applied to quantiles of measurement.

3.5 Materials and Methods

Consisting of 1,074,992 distinct observations and 41 attributes, the NSL-KDD dataset has been regarded as a viable benchmark for the study of intrusion detection amongst a variety of networks due to its exclusion of duplicate records for both its test and training sets and a distribution of observation of classification difficulties inversely proportional to its size [29]. For this study specifically, the labeled data was utilized in order to measure classification accuracy $N > 22{,}544$. Processing for this experimentation was utilized on a single system. All computation occurred using an Intel i9-9900K CPU at 3.60 GHz using 32 GB of DDR5 RAM.

Primary component analysis was conducted for the dataset prior to implementation, resulting in the selection of three attributes (Table 2). Normalization of all non-numeric fields was implemented in order to utilize k-means clustering. Elbow method implementation for denotes the optimal number of clusters for this dataset shown below (Fig. 5).

3.6 Data Observations and Extrapolation

The attributes derived from the principal component analysis were visualized in three-dimensions after k-means processing (Fig. 6). Three-dimensional visualization was also performed after cluster contraction with both a cluster view (Fig. 7) and an anomalous data point view (Fig. 8). Dimensionality reduction was also performed in the same manner for comparison and later spline implementation.

Table 2 Selected attributes from NSL-KDD dataset

Attribute	Description
rerror_rate	Percentage of connections that have "REJ" errors within a transmission
same_error_rate	Percentage of connections to the same service within a transmission
dst_host_same_srv_rate	Percentage of of connections to different services within a transmission

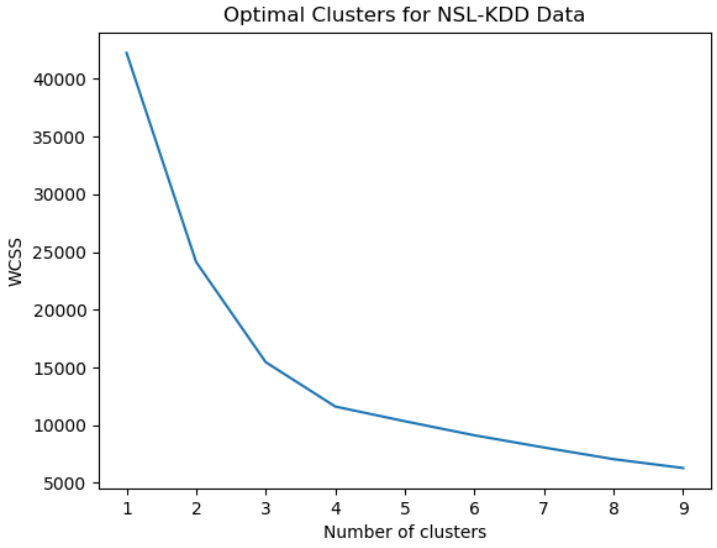

Fig. 5 The optimal number of clusters used for this dataset is four based on the summation of each cluster's distance between their centroids and their respective points (WCSS)

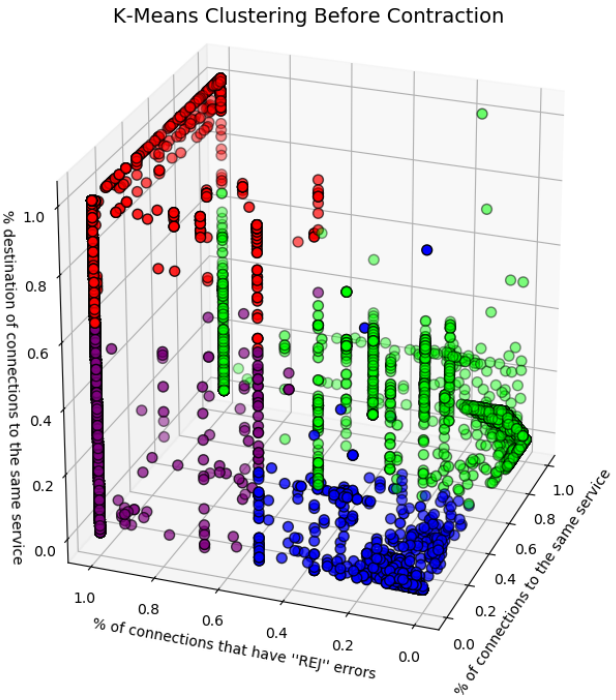

Fig.6 Visualization of NSL-KDD data before cluster contraction,. Each color represents an indivisual cluster

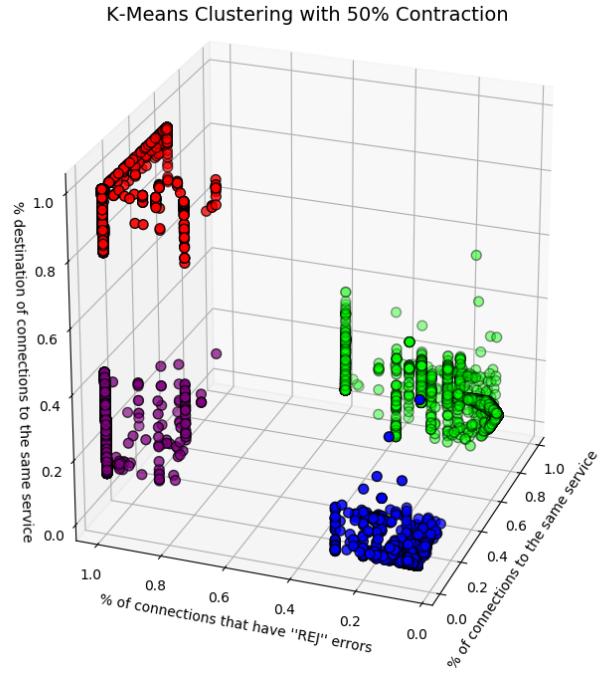

Fig. 7 Visualization of NSL-KDD data after cluster contraction. Each color represents an indivisual cluster

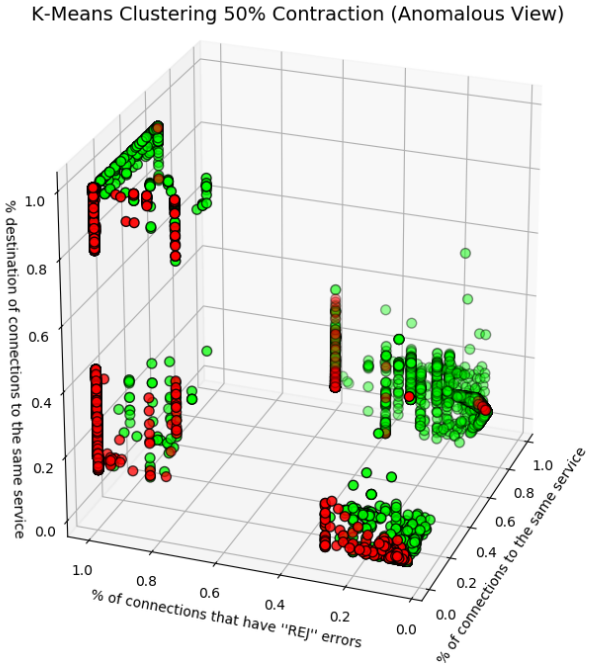

Fig. 6 Visualization of NSL-KDD data after cluster contraction. Red colored datapoints denote anamalous data and green denotes normal network traffic

3.7 Observations

Upon observing the three-dimensional visualizations, it can be noted that the majority of the pre-contraction data tends to cluster towards either the minimum or maximum for the range of each attribute (Fig 6). After contraction, the majority of anomalous data tends to form pillars or small groups that which reside toward the exterior of their respective cluster (Fig. 8)

3.8 Dimensionality Reduction

In order to visualize the data splines are implemented in, the previous three-dimensional depictions were reduced to second dimension visualizations. For the purpose of this experimentation, the *dst_host_same_srv_rate* attribute was removed (Fig. 8). Upon observing the two-dimensional visualization, it can be noted that a portion of the anomalous and normal data

now overlap after contraction (Fig. 9) which may lead to inaccuracies during classification (Fig. 10).

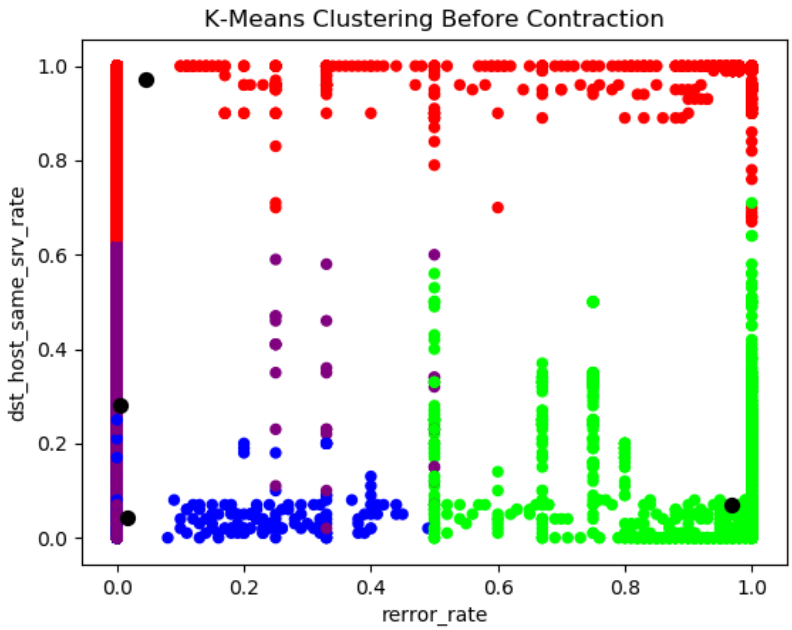

Fig. 7 Visualization of dynamic clustering before contraction. Black dots denote cluster centers. Each color is representatative of single cluster

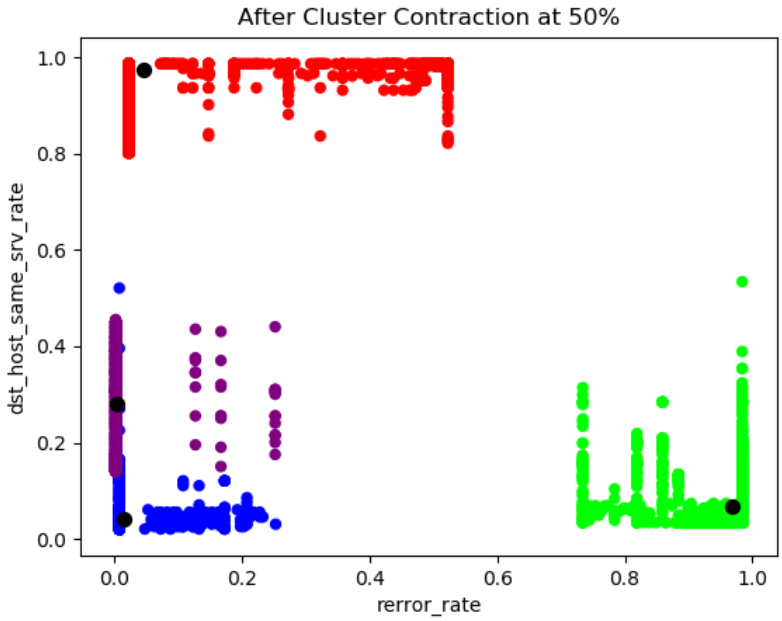

Fig. 8 Visualization of dynamic clustering after contraction. Black dots denote cluster centers. Each color is representatative of single cluster

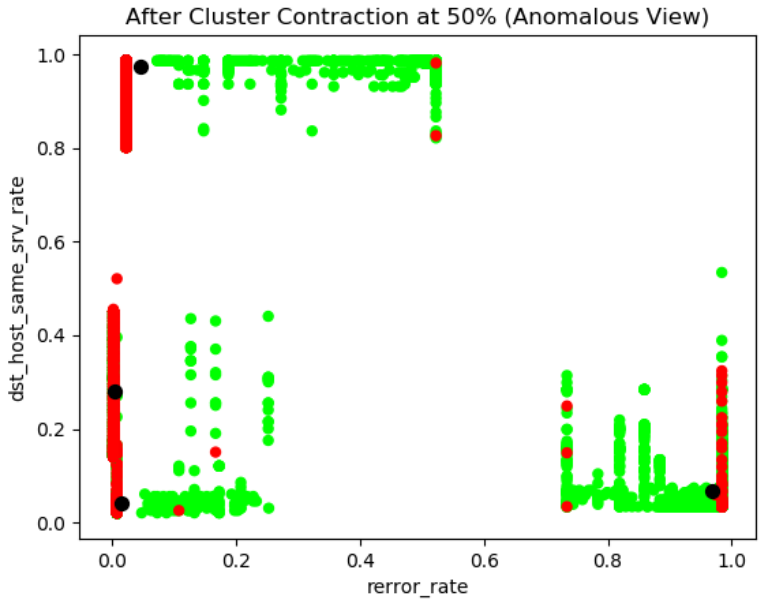

Fig. 9 Visualization of dynamic clustering after contraction. Black dots denote cluster centers. Red dots denote anamalous data. Green dots denote normal network traffic

3.9 Classification Implementation

K-means clustering was implemented using Python version 3.7.3's sklearn library [30]. Knot placement and spline application was implemented using R standard libraries in an RStudio environment. For run-time analysis, knot placement and spline application was implemented with Python using the stats models package. Splines were fit to the data based on a 20:80 test-train ratio of the NSL-KDD dataset. Accuracy readings for each spline were derived via confusion matrix analysis where classification of a malicious attack had a $p > 0.5$. Implementation protocol for spline-based classification can be seen in Algorithm 1 shown below. A range of knots spaced equidistant from one another was utilized for each calculation in order to understand their effects on classification accuracy and to identify an optimal number of knots for this dataset (Table 3). A logistic regression, and several support vector machines (SVM) using differing kernels were performed for comparison of classification accuracy (Table 4). The run time of the top performer from this comparison was then compared to KFC using a scaling number of observations.

Algorithm 1 Knot Flow Classification Algorithm

Algorithm 1 Knot Flow Classification

Result: Classification
initialization **while** *Means are variable* **do**
    k-means clustering assignment;
    Calculate Euclidean distance from each observation to each cluster center.
    Assign each observation to its closest cluster center.
    k-means clustering update;
    Calculate new means of the observations for the newly formed clusters.
    else
        cluster contraction;
        Reduce the distance between observations and their assigned cluster center by a predefined percentage.
        knot placement;
        Define equidistant locations for knot placement.
        spline placement;
        Calculate piece-wise polynomial functions between each knot in accordance to the probability of classification.
        Interpolate knots with calculated functions.
end

3.10 Classification Results and Discussion

Confusion matrix analysis denoted a classification accuracy greater than 73 percent for all splines. Splines utilizing three or more knots held a average accuracy of 79.37, as well as similar false negative and false positive counts (Table 4). The logistic regression scored an accuracy of 73.43 percent. The Naive Bayes classification scored an accuracy of 74.55 percent. The SVMs scored accuracy comparable to that of the splines with an approximate average accuracy of 78.76, with comparable false negative and false positive counts (Table 4). Accuracy findings for spline-based classification demonstrate nominal findings for the optimal number of knots. As demonstrated in Table 2, increases in accuracy stagnate for any spline with three or greater knots. False positive and false negative rates also remained relatively stable for this subset of splines, with an average false positive rate of 10.99 percent and an average false negative rate of 27.87 percent. Two of the biggest points of contention for this experimentation are the false positive and negative rates. As noted previously in this text, it is likely that these misclassifications are due to the dimensionality reduction needed to implement spline-based classification. In comparison with Fig. 8 and Fig 9, we can see that the splines correctly classify the anomalous data but, due to the overlapping data from the contraction, misclassify the underlying normal data. Run-time analysis demonstrated that KFC is marginally slower than an SVM when working with data consisting of less than 14,000 observations (Fig. 8). However, when working with data sets larger than 14,000 observations it outpaces the SVM. This out-pacing occurs due to the fact the KFC does not need calculate a hyper-plane between all of the test observations for to make a prediction and instead uses the newly formulated knot-horizon as a guide.

Table 3 Confusion matrix analysis and common classification techniques on pre-clustered data (N=4508)

Number of Knots	True Positive	True Negative	False Positive	False Negative	Accuracy
9	1500	2141	467	400	80.77%
8	1482	2171	437	418	81.03%
7	1466	2201	407	434	81.34%
6	1482	2171	473	418	81.03%
5	1553	2090	518	347	80.14%
4	1545	2101	507	355	80.08%
3	1453	2223	385	447	81.54%
2	1558	2061	547	342	80.513%
Logistic Regression	1460	1850	99	1099	78.42%
Naive-Bayes Classification	1832	1546	132	999	74.92%
Support Vector Machine Classification (Sigmoid Kernel)	1828	1648	146	887	77.09%
Support Vector Machine Classification (Linear Kernel)	1731	1818	221	739	78.71%
Support Vector Machine Classification (RBF Kernel)	1686	1954	286	583	80.73%

Table 4 Confusion matrix analysis and common classification techniques on clustered data (N=4508)

Number of Knots	True Positive	True Negative	False Positive	False Negative	Accuracy
9	1851	1728	214	716	79.37%
8	1853	1728	214	714	78.95%
7	1851	1726	212	716	79.40%
6	1851	1730	215	714	79.40%
5	1853	1730	215	714	79.41%
4	1853	1723	219	714	79.20%
3	1853	1721	221	714	79.26%
2	1846	1727	215	1078	73.43%
Logistic Regression	1460	1850	99	1099	73.43%
Naive-Bayes Classification	1808	1553	104	1734	74.55%
Support Vector Machine Classification (Sigmoid Kernel)	1724	1816	206	763	78.51%
Support Vector Machine Classification (Linear Kernel)	1701	1848	212	748	78.71%
Support Vector Machine Classification (RBF Kernel)	1728	1837	217	727	79.06%

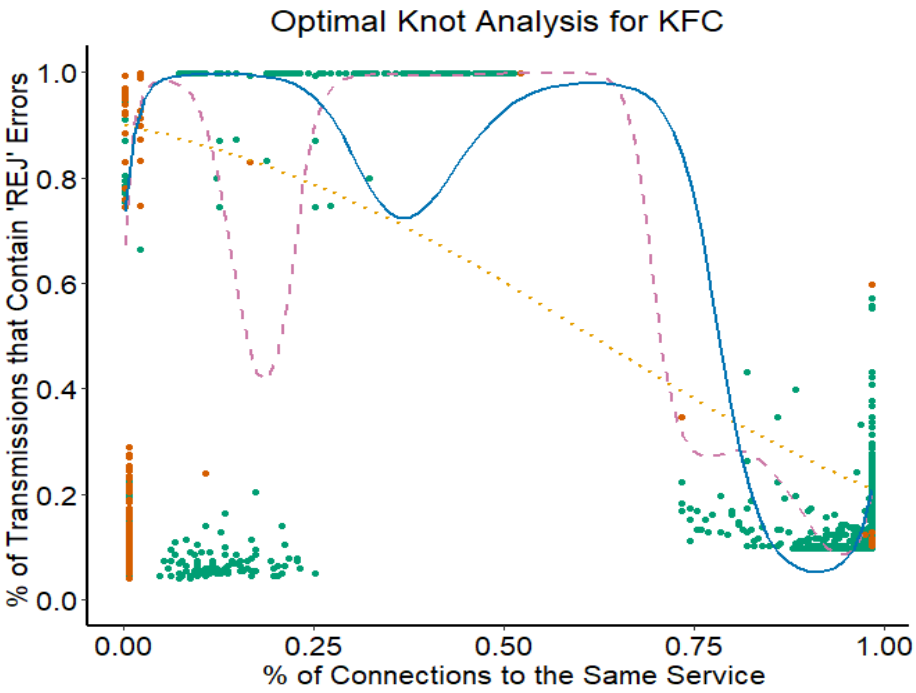

Fig. 10 Visualization of spline-based classifcation, using 3 knots (pink-dashed line) and 9 knots (blue solid line) in comparison to a logistic regression (yellow dotted line). Anamalous data points are denoted in orange and normal network traffic in green

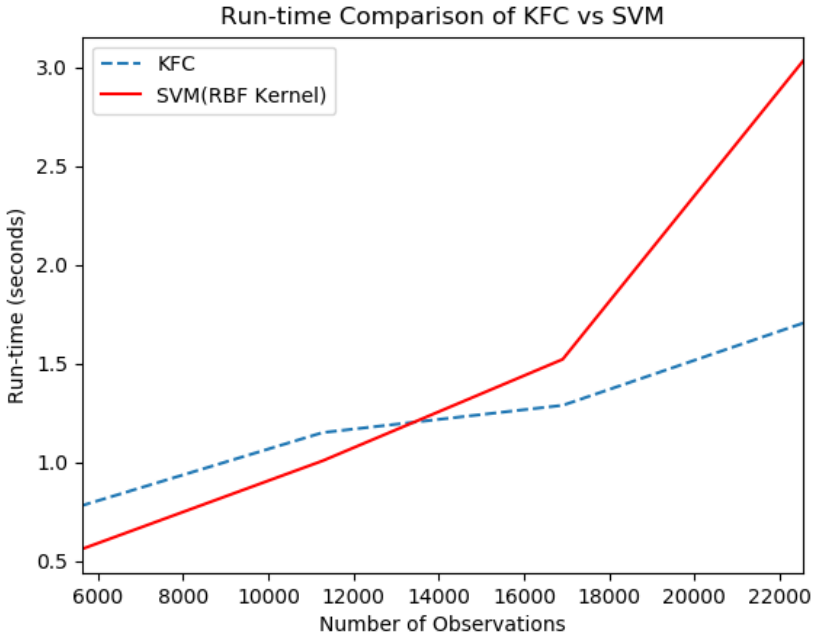

Fig. 11 Run-time compoarion of KFC to an SVM using RBF kernel. Runtime includes, loading of data, contraction of points and classifcation for both KFC and SVM

3.11 Classification Conclusion

Spline based intrusion detection demonstrates viable results when classifying anomalous network data. For this dataset, splines consisting of three knots a robust conduit for classification, allowing for a decreased amount of computation time when processing data. However, spline based dynamic clustering does have a major point of contention due to the need for dimensional reduction. In order to be used in a safety critical system within IoT, this point will be rectified in future research.

3.12 Chapter References

[2] Z. Hassan, H. Ali and M. Badawy, "Internet of Things (IoT): Definitions, Challenges, and Recent Research Directions," International Journal of Computer Applications, vol. 128, pp. 975-8887, 2105.

[8] H. Liao, C. Lin, Y. Lin and K. Tung, "Intrusion detection system: A comprehensive review," *Journal of Network and Computer Applications,* vol. 36, no. 1, pp. 16-24, 2013.

[9] M. Hasan, M. Islam, M. A. Zarif and M. Hashen, "Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches," *Internet of Things,* vol. 7, 2019.

[10] D. A. Schmidt, M. S. Khan and B. Bennett, "Spline Based Intrusion Detection in Vehicular Ad Hoc Networks (VANET)," *CoRR,* 2019.

[13] J. A. Haritigan, E. Bagheri, W. Lu and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," *IEEE Symposium on Computational Intelligence for Security and Defense Applications,* pp. 1-6, 2009.

[15] S. Sharma and K. A., "A survey on Intrusion Detection Systems and Honeypot based proactive security mechanisms in VANETs and VANET cloud," *Vehicular Communications,* vol. 12, pp. 138-164, 2018.

[18] L. L. Shumaker, Spline Functions : basic theory, Cambridge: Cambridge Printing House, 1981.

[26] R. Polikar, "Ensemble based systems in decision making," IEEE Circuits and Systems Magazine, vol. 6, pp. 21-45, 2006.

[27] P. H. Eliers and B. D. Marx, "Splines, knots, and penalities," Wiley Interdisciplinary Reviews: Computational Statistics, 2010.

[28] S. Spiriti, R. Eubank, P. W. Smith and D. Young, "Knot Selection for least-squares and penalized splines," Journal of Statistical Computation and Simulation, 2013.

[29] M. Tavallaee, E. Bagheru, W. Lu and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," IEEE Symposium on Computational Intelligence for Security and Defense Applications, pp. 1-6, 2009.

[30] F. Pedregosa, G. Varoquax, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay, "Scikit-learn: Machine Learning in Python," Journal of Machine Learning, vol. 12, pp. 2825-2830, 2011.

CHAPTER 4. CONCLUSION AND FUTURE RESEARCH

5.1 Conclusion

The emergence of IoT and its subsequent expansion into vehicular ad-hoc networks (VANETs) has created opportunities for a variety of researchers, engineers, and city planners to create applications that provide autonomous vehicles with preventative and mitigative safety measures built around an intrusion detection system (IDS). Due to the practical and ethical constraints associated with conducting IDS experimentation, these applications will need to be tested and validated in a laboratory setting before they are to be used in real-life situations. If IDS-centered simulations are to find a practical use in the validation and verification of VANET data, approaches like those described in this thesis will need to be developed. Moreover, these approaches will also need to be tested with a high rate of data processing, the management of massive volumes of this data, and the maintenance of highly accurate classification in mind.

On account this need, the research described in this thesis developed an intrusion detection system for vehicular ad-hoc networks. In my current understanding, this is one of the first implementations of an ensemble classification technique using basis splines and dynamic clustering, not only in the domain of IoT and VANETs, but in the realm of machine learning. Although the experiments conducted in this thesis display an effective and efficient means of classification, additional research is needed to fully understand the limitations of Knot Flow Classification. Furthermore, the full implementation of the KFC distributed system is needed to further inquire about its performance in a more realistic VANET environment. Lastly, the use of a highly specific VANET data is needed to fully encompass these claims.

5.2 Future Research

Future research based on the findings of this thesis could include the use of thin-plate tensor products for managing data of a higher dimensionality [31] [32]. Additionally, an automated principal component analysis for this data could be used as a form of data pre-processing of VANET data within the KFC distributed system using MapReduce and parallel processing concepts [14]. A conceptual design for this can be seen below (Fig. 14).

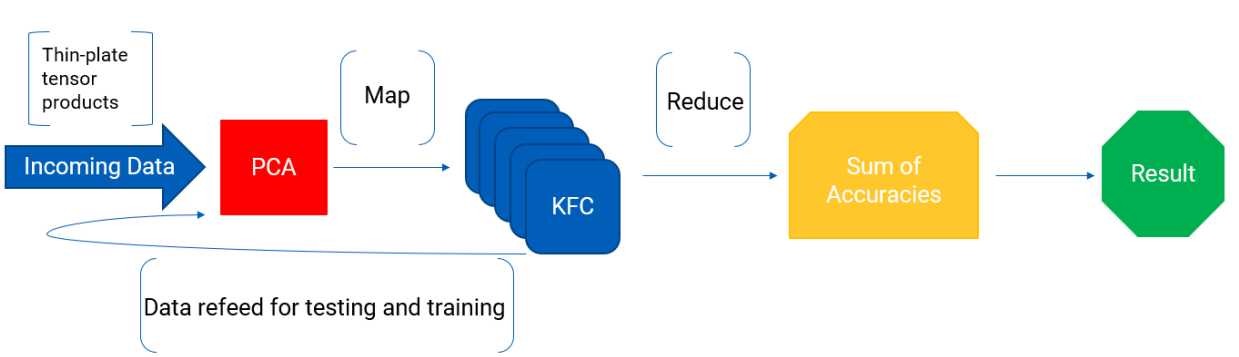

Fig. 12 Depiction of a distrbuted design concept for knot flow classification using thin-plate tensor products and MapReduce

5.2.1 Suggested Implementation

A tensor product is the block matrix result of two vectors. For example, if A is an $m \times n$ matrix and B is a $p \times q$ matrix then $A \otimes B$ is the $pm \times qn$ block matrix [31]. We can use these products to implement our primary component analysis by utilizing a matrix A to host our input data and a matrix B to host probabilities corresponding to each element in matrix A. The resulting attribute-probability pairs will allow us to select the features based on the probability of them denoting anomalous data. These tensor products can then be mapped across multiple instances of KFC and reduced to a sum of accuracies. These accuracies can then compare data point classifications resulting in an overall classification result.

5.2.2 Implications of Future Research

It is important note that the implications of this research extended to not only use in VANETs and the realm of IoT. If found to be applicable, the concepts introduced in this thesis can be expanded to many other fields of research where the management of large amounts of data and subsequent classification is needed. KFC could theoretically be applied to any type of data, with any number of attributes, to any number of observations (large or small). The only stipulation is that the attributes being processed are quantitative with any qualitative attribute being converted.

REFERENCES

[1] W. Liang, Z. Li, H. Zhang, S. Wang and R. Bie, "Vehicular Ad Hoc Networks: Architectures, Research Issues, Methodologies, Challenges, and Trends," *International Journal of Distributed Sensor Networks,* vol. 11, no. 8, 2015.

[2] Z. Hassan, H. Ali and M. Badawy, "Internet of Things (IoT): Definitions, Challenges, and Recent Research Directions," *International Journal of Computer Applications,* vol. 128, pp. 975-8887, 2105.

[3] K. M. Alam, M. Saini and A. E. Saddik, "Toward Social Internet of Vehicles: Concept, Architecture, and Applications," *IEEE Access,* vol. 3, pp. 343-357, 2015.

[4] Y. ,. e. a. Sun, "Security and Privacy in the Internet of Vehicles," *015 International Conference on Identification, Information, and Knowledge in the Internet of Things (IIKI),* pp. 115-121, 2015.

[5] M. Gerla, E. Lee, G. Pau and U. Lee, "Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds," *IEEE World Forum on Internet of Things (WF-IoT),* pp. 241-246, 2014.

[6] J. Leòn-Coca, D. Guitérrez, S. Toral, F. Barrero and N. Bessis, "Intelligent Transportation Systems and Wireless Access in Vehicular Environment Technology for Developing Smart Cities," *IEEE Access,* vol. 546, pp. 285-313, 2014.

[7] S. Karanki and M. S. Khan, "Secure multimedia delviery in vehciles using roadside infrastructure," *Vehicular Communications,* vol. 7, pp. 40-50, 2017.

[8] H. Liao, C. Lin, Y. Lin and K. Tung, "Intrusion detection system: A comprehensive review," *Journal of Network and Computer Applications,* vol. 36, no. 1, pp. 16-24, 2013.

[9] M. Hasan, M. Islam, M. A. Zarif and M. Hashen, "Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches," *Internet of Things,* vol. 7, 2019.

[10] D. A. Schmidt, M. S. Khan and B. Bennett, "Spline Based Intrusion Detection in Vehicular Ad Hoc Networks (VANET)," *CoRR,* 2019.

[11] E. A. Shams, A. Rizaner and A. H. Ulusoy, "Trust aware support vector machine intrusion detection and prevention system in vehicular ad hoc networks," *Computers & Security,* vol. 78, pp. 245-254, 2018.

[12] "NSL-KDD dataset," University of New Brunswick, [Online]. Available: https://www.unb.ca/cic/datasets/nsl.html. [Accessed 20 3 2020].

[13] J. A. Haritigan, E. Bagheri, W. Lu and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," *IEEE Symposium on Computational Intelligence for Security and Defense Applications,* pp. 1-6, 2009.

[14] A. McNabb, J. Lund and K. Seppi, "Mrs. Map Reduce for Scientific Computing in Python," *2012 SC Companion: High Performance Computing, Networking Storage and Analysis, Salt Lake City, UT,,* pp. 600-608, 2012.

[15] S. Sharma and K. A., "A survey on Intrusion Detection Systems and Honeypot based proactive security mechanisms in VANETs and VANET cloud," *Vehicular Communications,* vol. 12, pp. 138-164, 2018.

[16] A. Ashok, M. Sovindarasu and J. Wang, "Cyber-physical attack-resilient wide-area monitoing, protection, and control of the power grid," *IEEE Proceedings,* 2017.

[17] F. Sakiz and S. Sen, "A survey of attacks and detection mechanisms on intelligent transportation systems: VANETs and IoV," *Ad Hoc Networks,* vol. 61, pp. 33-50, 2017.

[18] L. L. Shumaker, Spline Functions : basic theory, Cambridge: Cambridge Printing House, 1981.

[19] "Spline Curves," Clemson University, [Online]. Available: https://people.cs.clemson.edu/~dhouse/courses. [Accessed 10 10 2018].

[20] S. Mukkamala, A. Sung, A. Ajith and V. Ramos, "Intrusion Detection Using Adaptive Regression Splines," *Enterprise Information Systems,* vol. 6, pp. 211-218, 2006.

[21] H. B. Lu and Q. L. Wang, "Intrusion detection based on spline neural network," *Jisuanji Yingyong Yanjiu Application Research of Computers,* vol. 26, no. 2, pp. 448-450, 2009.

[22] "Spline Approximation of Fuctions and Data," University of Oslo, [Online]. Available: https://www.uio.no/studier/emner/matnat/ifi/nedlagte-emner/INF-MAT5340/v05/undervisningsmateriale/kap5-new.pdf . [Accessed 28 12 2018].

[23] U. o. Houston, "Interpolation By Splines," [Online]. Available: https://www.math.uh.edu/~jingqiu/math4364/spline.pdf . [Accessed 28 10 2018].

[24] K. I. Joy, "Definition of B-Spline Curve," University of California Davis, [Online]. Available: https://www.cs.unc.edu/~dm/UNC/COMP258. [Accessed 1 1 2018].

[25] G. James, T. Hastie and R. Tibshirani, An Introduction to Statistical Learning with Applications in R, New York: Springer, 2013.

[26] R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits and Systems Magazine,* vol. 6, pp. 21-45, 2006.

[27] P. H. Eliers and B. D. Marx, "Splines, knots, and penalities," *Wiley Interdisciplinary Reviews: Computational Statistics,* 2010.

[28] S. Spiriti, R. Eubank, P. W. Smith and D. Young, "Knot Selection for least-squares and penalized splines," *Journal of Statistical Computation and Simulation,* 2013.

[29] M. Tavallaee, E. Bagheru, W. Lu and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," *IEEE Symposium on Computational Intelligence for Security and Defense Applications,* pp. 1-6, 2009.

[30] F. Pedregosa, G. Varoquax, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning,* vol. 12, pp. 2825-2830, 2011.

[31] G. Wahba, "Multivariate Model Building With Additive Interaction and Tensor Product Thin Plate Splines," *Curves and Surfaces,* pp. 491-504, 1991.

[32] P. A. Hancock and M. Hutchinson, "Spatial interpolation of large climate data sets using bivariate thin plate smoothing splines," *Environmental Modeling & Software,* vol. 21, no. 12, pp. 1684-1694, 2006.

VITA

DAVID ALLAN SCHMIDT

Education: M.S. Computer and Information Sciences,
Applied Computer Science,
East Tennessee State University,
Johnson City, Tennessee, 2020

B.S. Molecular Biology, University of Tennessee,
Knoxville, Tennessee, 2015

Public Schools, Knoxville, Tennessee

Professional Experience: Graduate Teaching Associate, East Tennessee State University,
Department of Computing,
Johnson City, Tennessee, 2019-Current

Graduate Teaching Assistant, East Tennessee State University,
Department of Computing, Johnson City,
Tennessee, 2018-2020

Associate Researcher, University of Tennessee,
Knoxville Tennessee, 2013-2015

Publications: Schmidt DA, Khan MS, Bennett BT.
Spline Based Intrusion Detection in Vehicular Ad Hoc Networks (VANET)
CoRR2019; abs/1903.08018.

Schmidt, DA, Khan, MS, Bennett, BT.
Spline-based intrusion detection for VANET utilizing knot

flow classification.

Internet Technology Letters. 2020;e155.

https://doi.org/10.1002/itl2.155

Honors and Awards:

Epsilon Pi Upsilon Honor Society Inductee, 2020

Sam Burke Outstanding Teaching Associate Award, 2020

Outstanding Graduate Student in Computing Award, 2020

East Tennessee State University "Illuminate" Nominee, 2020

East Tennessee State University Outstanding Thesis Nominee, 2020

Dean's List, 2018-2020

Nominee for Best Technical Editor University of Tennessee's Undergraduate Research Journal (Pursuit), 2015