Article

Exploiting Linear Support Vector Machine for Correlation-Based High Dimensional Data Classification in Wireless Sensor Networks

Lawrence Mwenda Muriira ^{1,*} , Zhiwei Zhao ¹ and Geyong Min ²

¹ College of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611700, China; zzw@uestc.edu.cn

² College of Engineering, Mathematics and Physical Sciences, University of Exeter, Exeter EX4 4QF, UK; g.min@exeter.ac.uk

* Correspondence: 201514060114@std.uestc.edu.cn; Tel.: +86-028-6183-1670

Received: 23 July 2018; Accepted: 22 August 2018; Published: 28 August 2018

Abstract: Linear Support Vector Machine (LSVM) has proven to be an effective approach for link classification in sensor networks. In this paper, we present a data-driven framework for reliable link classification that models Kernelized Linear Support Vector Machine (KLSVM) to produce stable and consistent results. KLSVM is a linear classifying technique that learns the “best” parameter settings. We investigated its application to model and capture two phenomena: High dimensional multi-category classification and Spatiotemporal data correlation in wireless sensor network (WSN). In addition, the technique also detects anomalies within the network. With the optimized selection of the linear kernel hyperparameters, the technique models high-dimensional data classification and the examined packet traces exhibit correlations between link features. Link features with Packet Reception Rate (PRR) greater than 50% show a high degree of negative correlation while the other sensor node observations show a moderate degree of positive correlation. The model gives a good visual intuition of the network behavior. The efficiency of the supervised learning technique is studied over real dataset obtained from a WSN testbed. To achieve that, we examined packet traces from the 802.15.4 network. The technique has a good performance on link quality estimation accuracy and a precise anomaly detection of sensor nodes within the network.

Keywords: linear support vector machine; linear kernel; correlation; high dimensional multi-category data classification; wireless sensor network

1. Introduction

The miniaturization of computing and sensing technologies enables the development of tiny, low-power, and inexpensive sensors, actuators, and controllers. These are sensing systems that typically interact closely with the physical world and are designed to perform a limited number of dedicated functions. Sensing is a technique used to gather information about a physical object or process, including the occurrence of events (for example, changes in state such as a drop in temperature or pressure). A device performing such a sensing task is called a sensor [1].

Sensors link the physical with the digital world by capturing and revealing real-world phenomena and converting these into a form that can be processed, stored, and acted upon [2]. Wireless Sensor networks (WSNs) are composed of cooperating sensor nodes that can perceive the environment to monitor physical phenomena and events of interest [3]. The dynamic environment makes the wireless communications vulnerable to distortion. The exposure of WSN to these external factors limits their performance due to their computing, storage and energy constraints.

The fast development of information from users and their applications; their networks and the cloud [4] have generated information diversification and complex relationship among information data [5]. Internet of Things that will connect the Internet with any kind of devices, living beings, and things will be the largest source of data in the near future. Wireless Sensor Networks are expected to have a key role in the realization of the future Internet of Things. Understanding the data deeply (the scale, structure, dimension, and correlation) guides the development of networks and information. Data correlation can solve the problems in data clustering, personal query and social network prediction [5,6].

Empirical WSN data correlations may exist in sensor network in three forms: correlation existing between sensor node observations on its history (temporal correlation), correlation existing between sensor node observation and neighboring nodes observation (spatial correlation), and correlation existing among attributes of sensor nodes (attribute correlation) [7]. Apart from the three basic types of data correlations mentioned, considering the temporal and spatial correlations together forms spatiotemporal correlations [8]. The combined correlation increases the detection rate of events on real dataset [9].

A major challenge for sensor nodes is sending sensed data to their sink in a reliable and energy efficient way, due to their limited resources in transmission, storage, computational memory and energy [10]. With the advancement in microelectromechanical systems (MEMS) technology, it is possible to deploy large scale WSNs, which introduce many data to be processed, transmitted and received. Transmitting all data back to a base station for processing and making inferences is impossible due to the sensor bandwidth constraints [11]. In addition, other challenges arise such as faulty node and outlier detection in target large-scale networks.

The utilization of machine learning seems to be one of the most convenient solutions for detecting anomaly in WSNs. Discovering an anomaly is a major principle to ensure a mundane functioning of WSNs [2]. Anomaly detection in WSNs is a consequential aspect of data analysis in order to identify data items which does not conform to an expected pattern in a dataset. It is paramount that the anomaly of sensor node is detected to obtain precise information to consequently make efficacious decisions by accumulated information.

For anomaly detection and utilization of data correlation for high dimensional classification in WSNs, there is a need for applying Support Vector Machine (SVM) with a kernel trick technique to mitigate such challenges. Some of the common types of kernel tricks applied are: Linear kernel, Polynomial Kernel, Radial Basic Function (RBF) kernel and Sigmoid Kernel [12]. These kernels are used to map dataset into high dimensionality feature space. In addition, the function of these kernels are to take dataset as input and transform it into the required form, with little computational cost even in very high-dimensional spaces. RBF kernel is the most popular kernel trick applied to non-linearly separable boundaries. Thus, the linear kernel works well if the datasets are linearly separable. However, the complexity of the RBF kernel grows with the size of the training. It is much easier to over-fit a complex model, because they have more hyperparameters to tune. This makes training of an RBF kernel SVM more computational costly and, furthermore, during prediction, the projection into “infinite” higher dimensional space where the data become linearly separable is more expensive than using LSVM [13,14].

To overcome this problem, we study a Kernelized Linear SVM technique for correlation-based high dimensional data classification and anomaly detection in WSNs. This approach is a linear classifying technique that learns the “best” parameter settings. The focus of our classification approach is not on generalization but on the accuracy of our classification for WSN’s link quality estimates, which are linearly separable. Our investigations show that the scattered plots of sensor node observations would tend to cluster around straight, non-horizontal lines and the sensed data are linearly correlated. This strategic technique and with a proposed data-driven framework for reliable link classification (see Figure 1) will provide insight into spatiotemporal correlations, high dimensional multi-category classification and anomaly detection, given an understanding of the distributive characteristic of WSNs.

Figure 1. Kernelized Linear Support Vector Machine (KLSVM) Model: the Data-driven framework for reliable link classification.

Prior studies of WSNs have observed that links have a wide range of PRR which can vary significantly overtime [15]. To determine whether the network behaves similarly or dynamically, we measured our packet reception rates on a 4×5 TelosB testbed. We borrowed an approach from [16] and used beacon-based measurement (BCM) to capture received beacon messages receptions/losses in bitmap denoted as “1” for reception and “0” for loss. From the bitmaps, we computed PRR, retransmission count and Conditional Probability Delivery Function (CPDF). We used PRR to measure link quality estimates and the estimates were classified in a similar approach used by Shu et al. [17] as: Faulty Link with 0% PRR, Very Bad Link with $PRR < 10\%$, Bad Link with $10\% \leq PRR < 45\%$, Intermediate Link with $45\% \leq PRR < 75\%$, Good Link with $75\% \leq PRR < 90\%$ or Very Good Link with $PRR \geq 90\%$.

The contributions of this paper are threefold. Firstly, we designed a data-driven framework for reliable link classification that models KLSVM to reveal two phenomena: high dimensional multi-category classification and spatiotemporal correlations in the sensor network. The design framework also detects anomalies caused by noises. Secondly, we studied the KLSVM model in the context of machine learning. We applied KLSVM technique to model high dimensional data classification, capture data correlation and detect anomalies. Finally, from empirical packet link traces collected from the WSN to our experimental evaluation, we show that the KLSVM technique modeled high dimensional multi-category classification over a real dataset. The approach captured temporal and spatial data correlation in 2D and 3D plots from the sensed data. We show KLSVM algorithm detects anomalies within the sensor network and models accordingly to more accurate diagnosis results. To the best of our knowledge, this is the first attempt to introduce KLSVM to experimentally model the two phenomena of spatiotemporal data correlation and high dimensional multi-category data classification based on link quality estimations in the field of WSNs. In addition, the technique detects anomalies within WSN in a single generic algorithm, paving the way to a sizable number of possible domains which can benefit from this technique.

This article provides in-depth insight into the KLSVM model and its application on a real dataset from a data classification and correlation perspective. The remainder of this paper is organized as follows. Section 2 is on related works. Section 3 is an in-depth description of the KLSVM model. Section 4 presents results and discussion of KLSVM. Finally, Section 5 concludes this paper.

2. Related Work

2.1. Preliminary

SVM algorithm is suited for extreme datasets separated by a Hyperplane. Thus, SVM can be defined as a frontier which best segregates two classes. If it does not have unoptimized decision boundary, it could result in greater misclassification on new data. Support vectors are defined as the data points that the margin pushes up against or points that are close to the opposing class. Therefore SVM algorithm implies that only these support vectors are essential, whereas other training examples are ignorable [18] as seen in Figure 2.

Figure 2. KLSVM—A correlation-based high-dimensional multi-category data classification model with a 3D hyperplane. The support vectors around the hyperplane are consequential training examples under SVM classification.

Let x^+ be the shortest distance to the closest positive point and x^- be the shortest distance closest to the negative point, then we have a margin M of the separating hyperplane which is the distance between x^+ and x^- . The line (decision boundary) that segregates the two classes is referred as the Hyperplane. SVMs are used in multidimensional dataset and the data are referred to as vectors. In LSVM, classes are linearly separable but in those cases that the data are not linearly separable, a function is used to transform the data to high dimensional space [19]. The process is computationally expensive and kernel trick is used to reduce the computational costs [20]. Therefore, such a function that takes inputs as vectors in the original space and returns the dot product of the vectors in the feature space is called a Kernel Function also referred as Kernel Trick [12].

Using a kernel function, one can apply the dot product between two vectors that every point is mapped into a high dimensional space via some transformation, and using it to transform a non-linear space into a linear space. Aforementioned, some of the popular kernel types are as follows:

- Polynomial Kernel of degree d corresponds to a particular degree d expansion of the features $K(a, b) = (1 + \sum_j a_j b_j)^d$. Data on the direction of a data point $x > 0$, the larger the values of x shows a corresponding increase on the Gram Matrix $K(X, X')$. Studies have shown polynomial kernels have been applied to overcome challenges in various fields, such as person re-identification problem by correctly matching person image from a set of gallery of person images in the field of machine learning [21], new malware detection in cyber-security [22], big data classification [23], facial recognition on multiple face images [24], human body movement and posture recognition in the medical field [25], facial emotion recognition [26] and classification of human brain images against mental health conditions in medical imaging [27].
- Radial Basic Function (RBF Kernel), also known as the Gaussian Similarity Kernel defined as $K(a, b) = \exp(-(a - b)^2 / 2\sigma^2)$. It results in high values near the point of x and falling off as a Gaussian with some spread σ as it moves away from point x . Here, the parameter σ can be

used to control over an under-fitting for σ chosen very large, all the data will look similar to any particular test point. This addition to the value R in Dual Optimization (discussed later in Section 3.6.2) which effectively controls the severeness of the penalty for misclassified points. RBF Kernel has been applied in different field of study such as detection of network intrusion in network security [28], enzyme discrimination in acidic and alkaline compound composition in biomedical engineering [29], emotion recognition from geometric facial features in digital image processing [30], stock market index predictions in the financial market [31], battery management system for optimized energy utility in electronic engineering [32], short-term wind power prediction in renewable energy [33], predicting traffic flow in smart city [34], load sharing and voltage compensation of micro-grids in electricity power distribution in electricity management [35] and data mining for regions susceptible landslide in China in the field of geographical information systems [36].

- Sigmoid Kernel $F(a, b) = \tanh(ca^T b + h)$, also known as Hyperbolic Tangent Similarity Function, transits from zero and moves in the direction of the x solution, as long as that data is sufficient in the direction of x . However, far in the direction of greater values, x gets high similarity values [14]. Recent studies have shown the application of the following fields of research, such as protein sequence classification in Biomedical research [37], network anomalies Detection for intelligent power substation [38], landmark recognition in digital image processing [39] and facial expressions recognition for videos [40].

Choosing the correct kernel type is a non-trivial task and may depend on specific tasks at hand. No matter which kernel is chosen, one needs to turn the kernel parameter to get good performance of its classifier [41]. A popular parameter tuning technique includes K-Fold Cross validation [42]. K-Fold Cross-Validation is used to evaluate the performance of the machine learning algorithm. Cross-Validation gives the most mileage on the training data and performance metric are averaged across K-Fold [17].

Advantages of using SVMs include [43]: They are effective in high dimension spaces [18]; SVMs use the subset of training points in the decision function on support vector; it is memory efficient; different kernel functions can be specified for various decision functions; and kernel functions can be added together to achieve even more complex hyperplane. SVMs also have some drawbacks such as, if the number of features is greater than the number of samples, the algorithm is likely to give poor performance [44]. SVMs also do not directly provide probability estimates, these are calculated using expensive techniques such as the K-Fold Cross Validation.

SVM algorithm has numerous application for example: medical imaging, regression modeling, image interpolation, medical classification tasks, financial industry time series prediction, financial analysis, application in neural networks for coding theory and practice, faulty diagnosis, page ranking and benchmarking, and objection recognition [45].

Recent studies have shown SVM is not only a classification technique but also an algorithm that can detect fault and anomalies [2], identify outliers [46], estimate and predict link quality [47,48], be used as an energy efficient routing method [43], and monitor and detect structural damage [42]. Trinh et al. [49] applied a data-driven hyperparameter optimization technique to detect anomaly in sensor networks using one-class support vector machines with radial basis functions kernelized. Gui et al. [42] used a three optimization-algorithm grid search, partial swarm optimization and Gaussian kernel function parameters for damage detection on architectural structures. When the number of features are too large SVM performs poorly. Ghaddar and Naoum-Sawaya [44] proposed an approach based on iteratively adjusting a bound on the l_1 -norm of the classifier vector to force the number of selected features to converge towards the desired maximum limit.

2.2. High-Dimensional Classification Techniques

Shu et al. [17] proposed a link quality estimation mechanism based on a multi-class classification SVM, modeled on two Kernel functions (radial basis and polynomial). Their model combined spatial correlation based data aggregation and opportunistic routing that achieved an improved network performance. Gholipour et al. [47] tackled the problem of network congestion and its improvement on WSN's throughput by using a multi-classification obtained from SVMs. They used a genetic algorithm to tune parameter in their simulations. Effective estimation for link quality guarantees reliable data transmission. The model estimates the current link quality accurately with a relative small number of trained packets. Salberg [50] used a linear SVM classifier to detect and classify objects in remote sensing images. The model was used to classify accurately the desired classes such as non-seal, adult seal and puppy seal.

2.3. Correlation Techniques in WSN

Data correlation is characterized among sensor nodes especially under densely deployments, affecting network's performance. Focus by researchers on data correlation in WSN has influenced designing of efficient routing protocols [51]. As more and more sensors are being deployed in the environment to provide sensing data to support various Internet of Things (IoT) applications, use of temporal and spatial correlations among different sensor data identifies the usefulness and correctness of each sensor data for different applications under IoT [52]. Utilization of temporal correlation of mobile sensing nodes on crowdsensing for fine-grained urban environmental monitoring has been achieved by fusing the sensory data with correlated environmental information [53]. According to Kumar and Kumar [54], their model built on Spatial and Temporal Data Correlation Algorithm for Data Aggregation overcame the challenge of data aggregation during flooding. With the dynamic characteristics of WSN, spatiotemporal data correlation approaches have been applied to data collection, aggregations, dissemination and network evaluation [9,55].

Similarly, studies have shown data correlation techniques can monitor detect fault and anomalies in WSN [7], predict link quality and energy efficient routing methods [51,54]. Densely deployed sensor nodes are prone to severe data redundancy due to the readings of neighboring nodes. High data redundancy leads to more energy consumption of the network. Large scale deployment causes congestion during transmission that leads to data losses and data retransmission is costly to WSN because of their limited energy budget [8]. Huang et al. [51] proposed a correlation aware opportunistic routing protocol that reduces redundancy by aggregating correlated data from selected forwarding nodes and the opportunistically forwarding the aggregated data to the sink. Here, we show how data correlation technique can detect faulty, anomalies and outliers within the WSN.

2.4. Anomaly Detection

Anomaly detection in networks and systems has enticed a large amount of attention recently. Erfani et al. [56] used a hybrid model to detect anomalies in large, high-dimensional datasets. Deep belief networks technique was applied for unsupervised training to extract features and while linear SVM was trained from the extracted features. However, Jedlinski and Jonak [57] used RBF-SVM for mechanical diagnosis to monitor vibration signal obtained from sensors on gearboxes. The Radial Basic Function was applied as the investigating method to enable early fault detection of gearboxes. An error-correcting output codes SVM is proposed for the multi-fault diagnosis of sensors. The author's [58] approach solved the sensor fault feature extraction and an online identification problem on sensors. Garcia-Font et al. [59] also applied RBF for the work on study detecting anomalies in WSN for smart city. In clinical research, Wang et al. [60] used SVM with three independent classifiers to diagnose and classify information of healthy or pathological brain detection. The authors aimed to build an automatic classification system of brain images in magnetic resonance imaging (MRI).

3. The KLSVM Model

3.1. Linear Classifier

SVM is a classification technique that splits data in the best possible way between two regions. A hyperplane is used to best split the data by fitting the maximum margin between support vectors in the classified data. Support vectors are the data points for training on the margin. The process of maximizing the margin is a Constrained Optimization Problem which can be solved by using the Lagrange Multiplier technique [61].

The Lagrange's Theorem states the following. Suppose f and g are functions of two variables that have continuous first partial derivative and $\nabla g \neq 0$ throughout a region of the xy -plane. If f has an extremum $f(x^{(i)}, y^{(i)})$ subject to the constraint $g(x, y) = 0$, then there is a real number λ such that

$$\nabla f(x^{(i)}, y^{(i)}) = \lambda \nabla g(x^{(i)}, y^{(i)}) \quad (1)$$

The number λ is called a Lagrange Multiplier.

LSVM is a common tool used for linear classifier which learns the “best” parameter setting. LSVM gives a stable decision boundary with the widest margin between support vectors. To explicitly maximize the decision boundary, it is important to optimize the margin classifier. Let w_i be the weight associated with each feature x_i and a constant term b , such that $b + w_1x_1 + w_2x_2 + \dots$

3.2. Computing the Margin Width

To define the margin, let us assume that all parameters get all the link data correctly. For the setting of the parameter, the decision boundary is invariance to its scaling. Thus, we remove the scale invariance by defining class +1 in some region, class -1 in another and making those regions as far apart as possible (see Figure 3). Then, we can define this margin explicitly in terms of the hyperplane. Since no data are inside the hyperplane, we define the margin as the distance between the two regions:

$$f(x) = wx' + b \quad (2a)$$

$$f(x) > +1 \text{ in class +1 region} \quad (2b)$$

$$f(x) < -1 \text{ in class -1 region} \quad (2c)$$

all passing through zero centre

$$f(x) = 0 \quad (2d)$$

Figure 3. KLSVM: A modeled decision boundary obtained from the real dataset. The hyperplane separates two regions of the Very Good Link and the Good Link data points. The plot shows that the proposed method proves the accurate performance of the classifier.

We define margin $M = \|x^+ - x^-\| = \|rw\|$ and vectors $w \in \mathbb{R}^{n+1}$ to be perpendicular to the boundaries $f(x) = -1$, $f(x) = +1$ and $f(x) = 0$ such that $w \cdot x + b = 0$ and $w \cdot x' + b = 0 \Rightarrow w(x' - x) = 0$ is Orthogonal.

Choosing x^- such that $f(x^-) = -1$; let x^+ be the closest points with $f(x^+) = +1$ defined as $x^+ = x^- + rw$ since w is orthogonal to the planes. These closest points on the margin also satisfy $w \cdot x^- + b = -1$ and $w \cdot x^+ + b = +1$. Since x^- is on the negative hyperplane and x^+ is on the positive hyperplane, their linear responses are -1 and $+1$, respectively, such that

$$w \cdot (x^- + rw) + b = +1 \quad (3a)$$

$$\Rightarrow r\|w\|^2 + w \cdot x^- + b = +1 \quad (3b)$$

$$\Rightarrow r\|w\|^2 = 1 = +1 \quad (3c)$$

$$r = \frac{2}{\|w\|^2} \quad (3d)$$

Since

$$M = \|rw\| = \frac{2}{\|w\|^2} \|w\| = \frac{2}{\sqrt{w^T w}} \quad (4)$$

3.3. Maximum Margin Classifier

We optimize our parameter since it is a constrained optimization problem to best fit our margin. We do that by getting all the data points correctly as the maximized margin equation subject to the constraints. Thus, all our data lay in specified regions on the correct side of the margin, as shown in Figure 3.

$$w^* = \arg \max_w \frac{2}{\sqrt{w^T w}} \quad (5)$$

Finding the value of w^* that maximizes $\|rw\|$, we can equivalently find the value w that minimizes w^* length.

3.3.1. The Primal Problem

The vector w_j^2 with the smallest length will also be the w^* with the largest inverse length. Similarly, minimizing squared length instead does not change the location of the minimizer, defined as

$$w^* = \arg \min_w \sum_j w_j^2 \quad (\text{The Primal Problem}) \quad (6)$$

subject to

$$\begin{aligned} y^{(i)} = +1 &\Rightarrow w \cdot x^{(i)} + b \geq +1 \\ y^{(i)} = -1 &\Rightarrow w \cdot x^{(i)} + b \leq -1 \end{aligned}$$

by reforming the maximization as a minimization of the sum of w_j^2 .

To enforce the data constraint, we have one constraint per data point. If $y^{(i)} = +1$, the linear response will be greater than $+1$. If $y^{(i)} = -1$, the linear response will be less than -1 .

3.3.2. Quadratic Program

The margin problem is an example of classic optimization problem called the Quadratic Program. We minimize the Quadratic Program of the parameters $\sum_j w_j^2$ in Equation (6) subject to a collection m linear constraint on the parameter at each data point that lies in the correct region. Framing this way makes easy to apply optimization algorithm designed for quadratic programs for reference, it is the formulation a Maximum Margin Classifier for Primal Problem.

It is convenient to compact these $w \cdot x^{(i)} + b \geq +1$ and $w \cdot x^{(i)} + b \leq -1$ constraints parameters to a single phrase that works for both positive and negative regions:

$$y^{(i)}(w \cdot x^{(i)} + b) \geq +1 \quad (7)$$

In Equation (7), $y^{(i)}$ is either +1 or −1. If it is +1, the linear response should also be positive and greater than one. Thus, the product $y^{(i)}(w \cdot x^{(i)} + b)$ will be the same. If $y = -1$, then the linear response should be negative and the product will be greater than one. Therefore, our linear response should be larger than one in magnitude.

3.4. Lagrangian Optimizer

Let $f(\theta)$ be our objective function and $g_i(\theta) \leq 0$ be our *Constraint Function*, where θ represent data w and b such that $\theta = (w, b)$ is defined as

$$f(\theta) = \arg \min_{w, b} \sum_j w_j^2 = w^* \quad (8a)$$

subject to

$$g_i(\theta) = 1 - y^{(i)}(w \cdot x^{(i)} + b) \leq 0 \quad (8b)$$

by minimizing the weights subject to the constraint that enforces the correctness model of our margin on each data point.

Then, we introduce a Lagrange Multiplier λ used to enforce the constraint θ jointly with a simple constraint set

$$\theta^* = \arg \min_{\theta} \max_{\lambda \geq 0} f(\theta) + \sum_i \lambda_i g_i(\theta) \quad (9)$$

Introducing the Lagrange Multiplier λ for each constraint $g_i(\theta)$ is used to enforce the constraint. The *Lagrangian* is given by optimization over the original parameters $\min_{\theta}$ and $\max_{\lambda \geq 0}$, where θ is to be minimized while λ is to be maximized. There is also a simple constraint on $\max_{\lambda \geq 0}$ to be non-negative, so it is easy to initiate the values θ and λ together by gradient descent steps

$$g_i(\theta) \leq 0 : \lambda_i = 0 \quad (10a)$$

$$g_i(\theta) > 0 : \lambda_i \rightarrow +\infty \quad (10b)$$

consider optimizing over λ_i for any fixed data.

3.4.1. KKT Complementary Slackness

If the constraint $g_i(\theta)$ is satisfied, then $g_i(\theta)$ is negative and the largest obtainable is zero by setting $\lambda_i = 0$. If the constraint is not satisfied, $g_i(\theta)$ is positive, λ will increase, and then data will have to change to decrease $g_i(\theta)$ for that constraint. Any optimum to the original problem will be a saddle point of the new Lagrangian and vice versa.

$$\lambda_i > 0 \Rightarrow g_i(\theta) = 0 \quad (11)$$

3.4.2. Optimization of the Lagrange Multiplier

The *Lagrangian* can be optimized by enforcing inequality constraint, defined as

$$w^* = \arg \min_w \max_{\lambda \geq 0} \frac{1}{2} \sum_j w_j^2 + \sum_i \lambda_i (1 - y^{(i)}(w \cdot x^{(i)} + b)) \quad (12)$$

for $\theta > 0$ only on the margin.

Fixing λ can solve directly for w and b in terms of λ . This unconstrained quadratic function takes their derivatives and sets them to zero. It gives the optimal w^* to be a linear combination of the data, defined as

$$w^* = \sum_i \lambda_i y^{(i)} x^{(i)} \quad (13)$$

and, since any support vector has $y - wx + b$,

$$b = \frac{1}{N_{sv}} \sum_{i \in sv} (y^{(i)} - w \cdot x^{(i)}) \quad (14)$$

$\frac{1}{N_{sv}}$ averages the equation for numerical stability. Since λ_i is zero for non-support vector points $y^{(i)} x^{(i)}$, the $\max_{\lambda \leq 0}$ boundary in Equation (12) and solution w^* in Equation (13) depend only on the support vectors. Solve for b by plugging the margin of the hyperplane in its equation for the support vectors sv , by using solution w^* to write solely in terms of θ as

$$\max_{\lambda \leq 0} \sum_i [\lambda_i - \frac{1}{2} \sum_j \lambda_i \lambda_j y^{(i)} y^{(j)} (x^{(i)} \cdot x^{(j)})] \quad (15)$$

subject to $\sum_i \lambda_i y^{(i)} = 0$ since its a derivative with respect to $b = 0$.

3.5. Dual Form

The optimal value w in terms of λ_i is plugged in to get an optimization solely over λ . The resulting problem is regarded as the Lagrange Dual form of the original problem. Plugging in w^* of Equation (12) and rearranging it, we find that the dual is given by $\lambda_i - \frac{1}{2} \sum_j \lambda_i \lambda_j y^{(i)} y^{(j)} (x^{(i)} \cdot x^{(j)})$ of Equation (15) as a minimization over $\lambda \leq 0$. Enforcing stationary condition on b , the derivative with respect to $b = 0$, since the original equation was linear with b , then $\sum_i \lambda_i y^{(i)} = 0$ becomes a constraint. Notice Equation (15) is a quadratic function in λ with single linear constraint, so it is a quadratic program that optimizes m variables with $1 + m$ simple constraint of the objective function has m^2 dot products. The Lagrangian Dual is always lower bound on the original primal problem of the minimization over θ .

Quadratic problems such as these have a property called strong duality which states the value of this optimization over λ will be the same as the primal problem. This dual forms are used when m the number of data points is much smaller than the number of features N . Notice that in Equation (15) our optimization is now λ which is m evaluation of the objective of $\lambda_i - \frac{1}{2} \sum_j \lambda_i \lambda_j y^{(i)} y^{(j)} (x^{(i)} \cdot x^{(j)})$ is then $\frac{1}{m^2}$ and optimizing is usually between quadratic and cubic m depending on the solver used, then optimization tolerance.

3.6. Linearly Non-Separable Data

For not linearly separable data, the margin constraints cannot be satisfied for any values of w and b ; the large margin principle $\min_w \sum_j w_j^2$ for separable data suggests we should use a model with small parameters. However, if we are forced to have a non-zero error $\min_w \sum_j J(y^{(i)}, w \cdot x^{(i)} + b)$, we should trade off.

3.6.1. Slack Variable

The error that results from our prediction is caused by some of our solutions allowing some of our data points to violate the margin constraint. Thus, a Soft Margin is assigned a cost, for example, the distance by which they violated the constraints scale by a factor R . If the factor R is chosen to be very large, it pays a lot of attention to make sure no data violate the constraints margin if possible.

On the other hand, if R is very small, maximize the margin for most data to allow some of them to violate it. Figure 4 shows we do so by adding the so called Slack Variables $\epsilon^{(i)}$ one for each data points, defined as

$$w^* = \arg \min_{w, \epsilon} \sum_j w_j^2 + R \sum_i \epsilon^{(i)} \quad (16)$$

subject to $y^{(i)}(w^T x^{(i)} + b) \geq +1 - \epsilon^{(i)}$ violating margin by $\epsilon^{(i)} \geq 0$.

$\epsilon^{(i)}$ measures the amount of data point i violating the margin constraint. $\epsilon^{(i)}$ is always non-negative or zero if the constraints are satisfied and adds a penalty to epsilon to our objective function balancing the margin in terms of margin squared w_j^2 with the amount of slack $R \sum_i \epsilon^{(i)}$. Notice the formulation of w^* in Equation (16) remains a quadratic program; its quadratic objective in w and ϵ subject to linear constraint $y^{(i)}(w^T x^{(i)} + b) \geq +1 - \epsilon^{(i)}$. For any weight w , we can choose ϵ to satisfy constraints by writing ϵ^* as a function J and optimizing directly. For any weight vector w , we can always choose a value ϵ to minimize its term and satisfy constraints. It means, first, we can always initiate a solution w , ϵ and b to something that satisfies the constraints even if it does not minimize the objective. Secondly, the optimal value ϵ given w is easy to select if data point i is not satisfied, the margin with $\epsilon^{(i)} = 0$, then the smallest value ϵ , enforces these inequalities to be true and will accept two sides to be equal.

Figure 4. KLSVM: Linearly non-separable data points between the *Very Good Link* and the *Good Link* regions. The retransmission count patterns of these data points are inconsistent with the other data points of their classes.

By choosing an optimal value for a given w , we can then optimize the resulting cost as function of J directly as

$$J_i = \max[0, 1 - y^{(i)}(w \cdot x^{(i)} + b)] \quad (17)$$

where ϵ^* written as a function of J can be optimized directly to

$$w^* = \arg \min_w \sum_j w_j^2 + R \sum_i J_i(y^{(i)}, w \cdot x^{(i)} + b) \quad (18)$$

We find that the cost of J is only non-zero for data points that do not satisfy the margin constraint and for those points equal linearly with their distance to the margin.

3.6.2. Standard Linear Classifier Optimization

For positive data point $w \cdot x + b \rightarrow +1$, the linear response is already greater than +1 and has no cost $J = 0$. On the other hand, if it is less than +1, the cost will increase linearly with its distance from the margin. This kind of loss is called Hinge Loss. Its analytical form J_i as shown in

Equation (17) is piecewise linear, and is either a zero or positive course that increases away from the margin. Our overall optimization in Equation (18) is then a margin term $\sum_j w_j^2$ plus our slack variable R times the Hinge Loss J_j and a balance between the margin term and slack variables, also defined as

$$w^* = \arg \min_w \frac{1}{R} \sum_j w_j^2 + \sum_i J_i(y^{(i)}, w \cdot x^{(i)} + b) \quad (19)$$

Dividing Equation (19) by R , we get that the optimal parameter minimizes the sum of the data term and hinge loss, that is, $\frac{1}{R}$ times an l_2 regularization terms of their weights. This equation forms as a standard linear classifier optimization, but the difference is the Hinge Loss and not regularizing the coefficient b .

We can optimize $\sum_j J_i(y^{(i)}, w \cdot x^{(i)} + b)$ in whatever manner we like such as any standard Stochastic Gradient algorithm from the linear classifier. If we take the dual of the soft margin quadratic program, we obtain a quadratic program similar to before over only the Lagrange multiplier λ .

$$\max_{0 \leq \lambda \leq R} \sum_i \lambda_i - \frac{1}{2} \sum \lambda_i \lambda_j y^{(i)} y^{(j)} (x^{(i)} \cdot x^{(j)}) \quad (20)$$

subject to $\sum_i \lambda_i y^{(i)} = 0$. The λ_i is bounded from above as well by R . Intuitively, the equation says that, if the data point violates the margin constraints, the margin will penalize the Lagrange multiplier λ_i of that data point. Then λ will increase until it is at most R times the violated distance, defined as

$$w^* = \sum_i \lambda_i y^{(i)} x^{(i)} \quad \forall \lambda \in (0, R) \quad (21)$$

Complimentary slackness tells that λ is non-zero only on data that are at the margin or on the wrong side, for positive data at any point where the linear response is less than equals to $+1$.

3.7. Gram Matrix

The dual form can be important when there are many features than data points, that is, $N \gg m$, a key thing to notice about dual form of the SVM is that quadratic program involves the features x only through their dot products. In other words, the coefficient of i and j interaction of the term between λ_i and λ_j is the inner product of the data point of $x^{(i)} \cdot x^{(j)}$.

Let's call this inner product K_{ij} , that is, $i^{th} j^{th}$ entry in the matrix K sometimes called the Gram Matrix. We can think of the quantity as measuring the similarity of two data points $x^{(i)}$ and $x^{(j)}$ through their dot product if and only if the vectors are in the same direction. It takes its maximum value and its orthogonal at zero; if they are on opposite directions, it has negative values.

3.8. Prediction

Interestingly, predictions also only involve dot products using our solutions w^* . We find that our prediction $\hat{y}$ for a new test point x is a linear combination of the dot products of x with the support vectors $x^{(i)}$ the points where λ_i is non-zero, defined as

$$\hat{y} = w^* \cdot x + b = \sum_i \lambda_i y^{(i)} \cdot x + b \quad (22)$$

Evaluating b is a bit more complicated but it is before any support vector with slack on λ , not equal to 0 or R will have a tight margin constraint that can be used to solve for b . Typically, b is kept updated when λ has been solved.

3.9. Kernel Function

In practice, linear functions may not correspond to any particular feature transform but often seems to perform well empirically. For extremely large data, however, kernels and SVMs are less common and LSVM with explicit features are more typical. For datasets $O(m^2)$ cost, working in dual form may be too much and Primal optimizing the Hinge Loss directly using Stochastic Gradient Descent is preferred [12].

3.9.1. Kernelizing Linear SVM

LSVM has a simplified perceptron of a linear weight w on the input features resulting in a linear decision boundary $f(x) = 0$ developed by a Lagrangian optimization in the form:

$$\max_{0 \leq \lambda \leq R} \sum_i \lambda_i - \frac{1}{2} \sum_j \lambda_i \lambda_j y^{(i)} y^{(j)} (x^{(i)} \cdot x^{(j)}) \quad (23)$$

subject to $\sum_i \lambda_i y^{(i)} = 0$. That leads to an equivalent dual formation process in which the objective function depends on the matrix $K_{ij} = x^{(i)} \cdot x^{(j)}$ on pairwise dot product called the Gram Matrix.

If our data are not linearly separable on the original feature x_1 , we can add quadratic deterministic features, for example, moving from one feature x_1 to two features $x_2 = (x_1^2)$ so that the data lie on a curve. In our new higher dimension space (see Figure 5), our data are more likely to be linearly separable. This affects the dual form of the sum using a feature transform $\Phi(x)$ $\hat{y}(x) = \text{sign}[w \cdot \Phi(x) + b]$. We transform $x^{(i)}$ and $x^{(j)}$ to find the new feature vectors, and then the dual form involves the dot products between these transformed vectors, defined as

$$\max_{0 \leq \lambda \leq R} \sum_i \lambda_i - \frac{1}{2} \sum_j \lambda_i \lambda_j y^{(i)} y^{(j)} \Phi(x^{(i)}) \Phi(x^{(j)})^T \quad (24)$$

subject to $\sum_i \lambda_i y^{(i)} = 0$.

Figure 5. KLSVM: A modeled correlation-based high dimensional multi-category classification. In addition, the results show one detected failed link and four captured outliers, with extremely inconsistent behavior.

Let us consider the polynomial features defined as

$$\Phi(x) = (1 \sqrt{2x_1}, \sqrt{2x_2} \dots x_1^2 x_2^2 \dots \sqrt{2x_1 x_2} \sqrt{2x_1 x_3} \dots) \quad (25a)$$

For the dual form, we need to compute inner product of two expanded feature vectors $\Phi(x^{(i)}) \Phi(x^{(j)})^T$ by denoting $x^{(i)}$ and $x^{(j)}$ to a and b respectively. Listing our features for both points, we compute them as

$$\Phi(a) = (1 \sqrt{2a_1} \sqrt{2a_2} \dots a_1^2 a_2^2 \dots \sqrt{2a_1 a_2} \sqrt{2a_1 a_3} \dots) \quad (25b)$$

$$\Phi(b) = (1\sqrt{2b_1}\sqrt{2b_2}\dots b_1^2b_2^2\dots\sqrt{2b_1b_2}\sqrt{2b_1b_3}\dots) \quad (25c)$$

We find that the dot product is the sum, defined as

$$\Phi(a)^T\Phi(b) = 1 + \sum_j 2a_jb_j + \sum_j a_j^2b_j^2 + \sum_j \sum_{k>j} 2a_ja_kb_jb_k + \dots \quad (25d)$$

If we manipulate the sum algebraically, we find that it is equivalent to a much simpler computation

$$\Phi(a)^T\Phi(b) = (1 + \sum_j a_jb_j)^2 = K(a, b) \quad (25e)$$

denoting the value $K(a, b)$ instead of actually linking our higher dimensional feature transform and then computing the inner product of those vectors, we could instead take the very simple non-linear function similarity (the kernel) and compute on the linear and the original number of features our expansion fee has; for example, quadratic features $\Phi(a)$ and $\Phi(b)$ have all squared features and computing their dot product in only $O(n)$ computations.

3.9.2. Mercer's Kernel

In addition, a non-linear function $K(X, X')$ satisfying a particular condition called Mercer's Condition can be viewed as corresponding to the dot product between transform vectors $\Phi(x)$ for some transformational fee:

$$\int_a \int_b K(a, b)g(a)g(b)dadb > 0 \quad (26)$$

Then, the similarly function $K(a, b) = \Phi(a) \cdot \Phi(b)$ is also referred to as Mercer's Kernel (condition). As a side note, this Mercer's condition is effectively condition on the Gram Matrix K as a positive definite structure for any possible dataset x , defined as $g^T \cdot K \cdot g \geq 0$.

For polynomial features, there is direct response between a particular K and a particular feature vector Φ , however for arbitrary similarity function K , it might be quite hard to find exactly how vector feature Φ correspond to that K . In fact, many useful kernel functions correspond to infinite dimensions of Φ vectors. Thus, using such a kernel is equivalent to a linear classifier with an infinite number of constructed feature yet it is not computational harder as long as K itself is easy to compute. Then, we can calculate the gram matrix $K(X, X')$ in $O(m^2)$ time in space and solve the resulting quadratic programming to quadratic cubic times.

4. Results and Discussion

4.1. Data Description

We considered a dataset gathered from a WSN deployment at the New Generation Mobile Internet Research Laboratory (NGMI) in the University of Electronic Science and Technology of China. The results are from a 20-sensor-node testbed on the laboratory's ceiling. Figure 6 shows the sensor deployment in the laboratory. The nodes have identification numbers, *Node0* as the root node and the other nodes follow a sequential order (*Node1*, *Node2*, ..., *Node20*). The nodes in this experiment run on TinyOS and use TelosB sensor nodes.

We applied the BCM approach that uses the beacon receptions to provide historical statistics. To determine the actual packet traces results, two tunable modeling parameters were considered: window size W , representing the number of packets used in each single input, and the interval-packet interval I , representing the time interval between packets. The window size corresponds to the amount of historical information required by our experiment to estimate the link quality in our prediction. We programmed nodes on the meshed network topology testbed to broadcast 400,000 packets with an inter-packet interval of 50 ms using 16 channels. Each node transmitting 1000 packets to a relaying node or to the root node. Timing is an important factor for beaconed packet distribution and we needed

a way that could concisely describe link behavior from the packet traces. At $I = 50$ ms, the WSN's links stabilized to a near-perfect quality for our experiment and consequently our data collection technique also converged to a stable state. To measure the link features, we considered the bitmaps to calculate CPDFs, PRRs and retransmission count. Henceforth, the term link feature interchanges with either of these other terms, sensed data or data variable.

Figure 6. NGMI's ceiling: Part of the 4×5 WSN testbed.

CPDF is the probability the next packet will be transmitted successfully given a number of consecutive packet successes or failures. CPDFs can give a good visual intuition of link behavior. We wanted to present link behavior between nodes as a single scalar value. To do so, we used Earth Mover's Distance (EMD) function [62] to compute the retransmission cost. EMD computes the distance between successful packet trace distribution and the failed packet trace distribution with a window size of $W = 1$. While distance is informative, as shown in Table 1, it is not sufficient to measure link quality. Thus, we computed PRR to distill a measure of packet delivery successes and failures within a link. Measuring link behavior and its effect on the network performance, we computed the retransmission count showing the distribution of the number of resumed transmission upon packet failures.

Table 1. Average retransmission cost.

Link Type	Number of Links	Percentage	Average Retransmission Cost ¹
Very Good Link	49	12.25%	0.1015077
Good Link	93	23.25%	0.309878
Intermediate Link	181	45.25%	0.729711
Bad Link	63	15.75%	1.914515
Very Bad Link	13	3.25%	63.272 ²
Faulty Link	1	0.25%	N/A

¹ the average retransmission cost in various link quality estimations category; and ² for very bad links it on average costs 63.272 to transmit a packet successfully compared to the other link observations.

4.2. KLSVM Model Implementation

Using Python's sklearn library, the support vector classifier (SVC) was computed with $C = 1$ as regularization parameter, and $\gamma = 1$ and *decision_function_shape* = 'ovr' as the one-versus-all multi-category classification technique for the linear kernel. For our implementation design (see Figure 7), we used *model.fit()* as an effective method for inputting all the labels (link quality estimates), fitting linear feature data and training the data. The results returned are the optimal values for the parameters. The method *model.predict()* was also used as our prediction model that functioned as an extension to the generic algorithm.

Figure 7. KLSVM's training and prediction model. The inputs are the pre-processed packet traces.

4.3. High-Dimensional Multi-Category Classification

KLSVM is a supervised learning multi-category classification technique. It learns a linear kernel function from training data consisting of pairs of pre-processed packet traces, as input features, and link quality estimates, as categorical output. The linear kernel function is used to predict a class label of any valid input feature. Interestingly, within the research field of multi-category classification, there exist two types of existing methods for handling multi-category classification problem that can be distinguished: pairwise classifiers and one-versus-all approach. Whereas the former seeks to solve a series of binary classifications, the latter considers all the classes simultaneously. In our technique, we considered one-versus-all because we required training X distinct binary classifiers to separate one class from all others and each binary classifier uses all training samples. For the pairwise approach, there are $X(X - 1)/2$ binary classifiers to be trained with one for each pair of classes. Compared to the one-versus-all approach, the number of classifiers is much larger for the pairwise approach but each one involves only a subsample of the training data and thus is easier to train.

The remarkable recent development of sense computing, powerful state-of-the-art technology and massive storage technology has allowed scientists to collect data of unprecedented size and complexity. When the dimensionality of the input feature space is too large, things become complicated, for instance, if the raw packet traces were used for training. Another difficulty of high dimensional classification is caused by the existence of many noise features such as outliers that do not contribute to the reduction of classification error. Although individually each extracted feature from the examined packet traces can be estimated accurately, aggregated estimation error over the raw packet traces (if they were to be used as features) can be very large and this could significantly increase the misclassification rate, which places more emphasis on misclassification rates rather than the accuracy of estimated categorical output. Even though in high dimensional classification, the focus is much more about the misclassification rate instead of the accuracy of the estimated parameters. This characteristic makes the LSVM incline to be sensitive to training noisy data. When there exist outliers within their own classes, LSVM classifier inclines to be vigorously affected by such far away points due to the unboundedness of the hinge loss. Figure 8 shows, the directions found by the KLSVM technique puts much more weight on link features that provide large classification power.

Figure 8. KLSVM Model: A visual intuition of the network behavior.

Generally, linear prediction methods are likely to perform poorly unless the prediction vector $\hat{y}$ is sparse, that is, the effective number of selected features is small. This is due to, as earlier mentioned, the noise accumulation to a large extent as a high-dimensional problem. Classification technique using all features do not necessarily perform well due to the noise accumulation when estimating a large number of noise features. Dealing with high dimensionality and small sample size mitigates the “curse-of-dimensionality”. The “curse-of-dimensionality” refers to a phenomena that arises when analyzing and organizing the dataset in high-dimensional spaces with very large-dimensional settings that does not occur in low dimensions such as the 3D physical space.

4.4. Spatiotemporal Data Correlation

Since spatiotemporal correlations exist among sensor readings, thus correlation is fundamentally about understanding the direction and strength of the relationship between sensed data variables. Although correlation can be used with hypothesis testing [63], it does a number of other useful roles in this section, as follows:

- **Reliability:** A strong correlation between link features on a test means that they are consistent with measuring the same behavioral outcome. If all the link features on the test correlate well, then it is a reliable test.
- **Validity:** When developing a brand new test of intelligence, one could empirically want to demonstrate that the test correlates strongly with an existing measure of intelligence, measuring the same construct as the first intelligence test. A strong correlation between a new test and “Gold Standard” test, such as link quality estimations and their multi-category classification, means they are measuring the same construct and therefore the test is valid.
- **Prediction:** Experimentally, if link features are strongly correlated such as a drop in packet transmission count and increase in PRR, the next time PRR begins to drop, one can predict that link quality will deteriorate soon. When two data variables are correlated, when one changes, it can be *predicted* what happens with the other.
- **Verification:** It is for theory verification or testing a theory, if a link feature is causing changes in another link feature, then they are strongly correlated. If this theory is correct, then an outcome x should happen. If the link feature turns out to be uncorrelated or are correlated in the wrong direction, then it is concluded that the link feature is certainly not causing changes in others, thus it is noted that the theory is not correct.

Data correlation is examined in our study between three link features, PRR, retransmission count and CPDF (measured in percentage) from the packet traces captured from the 802.15.4 testbed.

As earlier seen in Figure 5, the change in one link's feature, reacted by an equivalent unit change in the other link's feature directly or indirectly. In other scenarios, variables are said to be uncorrelated in amount when one data variable does not show any movement in another data variable in a specific direction.

Figure 9 shows data correlation can be positive or negative with the two link features moving in the same direction. An increase in one link feature results to a corresponding increase in the other link feature and vice versa. Then, the link features are considered to be positively correlated. On the contrary, when the two link features move in the different direction in such a way that an increase in one link feature will result in the decrease in the other link feature and vice versa, it is empirical said to be negatively correlated.

Figure 9. KLSVM: Sensor node observations with PRR greater than 50% have a high degree of negative correlation and a moderate degree of positive correlation for the rest of the observations. These observed links with PRR less than 50% show inconsistent pattern of their retransmission counts, thus the cause of a moderate degree of positive correlation.

4.5. Detected Anomalies

4.5.1. Outliers

Outliers indicate abnormal sensing conditions and their detection is a critical task in many safety-critical environments that sensors are applied. In our experiment, we observed that four links laid abnormally far away, clearly isolated and inconsistent with the pattern of the other links. Such outcomes could be due to the quality of sensed data that may have been affected by noise, error or inconsistent packet reception. Because outliers are one of the sources that greatly influence the quality of link for transmission, in this section, we provide an overview of the results of detected outliers. Table 2 gives a description of our findings.

Table 2. Diagnosis results of the outliers.

Tag ¹	PRR ²	RC ³	Chnl ³	TL ³	Description
1	352	311	16	Node 7 to Node 1	The link exhibits a busty traffic pattern with frequent long packet losses.
2	446	174	12	Node 5 to Node 3	The link exhibits a busty traffic pattern with frequent long packet losses. Notably, after the 780th transmission only 11 packets were delivered.
3	272	108	12	Node 7 to Node 3	The link exhibits a busty traffic pattern with frequent long packet losses. Notably, after the 780th transmission only 5 packets were delivered.
4	248	101	12	Node 6 to Node 3	The link exhibits a busty traffic pattern with frequent long packet losses. Notably, after the 779th transmission only 3 packets were delivered.

¹ In Figure 8 all the outliers are tagged sequentially. ² The outliers are in the range of “Bad Link” quality estimation. They are ignored because they are not reliable for end-to-end communication. ³ Abbreviations as used in the table: RC (Retransmission Count), Chnl (Channel) and TL (Transmission Link).

KLSVM approach prevented normal sensed data from being classified as outliers and kept the detection rate high and false alarm rate low. The technique identifies outliers from the sensed data measurements but not performed in real-time. The technique pays attention to spatial correlation of neighboring nodes, which makes the results of outliers accurate.

4.5.2. Faulty Link

As earlier seen, Figure 8 shows a detected link failure in the meshed network topology we used for data collection. In a meshed WSN topology, sensor nodes can communicate with numerous nodes within the network, guaranteeing redundancy against connection failures. Connection failures are common in sensing systems because of the inherent unreliability of the shared communication media. Besides, connection disappointments are often caused by hardware failure, software failure, communication failure, aging phenomenon or human error. Since information sent over a fizzled link is lost, the time between a failure and its detection is pivotal for a solid end-to-end communication in the Mesh WSN. Once the link failure is recognized, an effective route is chosen by the routing protocol and communication can proceed. In this way, to increase the unwavering quality of the Mesh WSN, the discovery time for link failures must be kept as low as can be expected under the circumstances. The link failure happened between Node 9 and Node 3 using channel 12. Our investigation reveals that Node 3 temporally lost power at the time Node 9 was establishing its connection. Thus, the detected faulty link was due to a hardware failure.

5. Conclusions

Generally, LSVM has the ability to minimize training error and maximize the margin over dataset, whose labels are included as supplemental variables in the optimization problem. While other classification methods focus on conditional probabilities, LSVM targets estimating the decision boundary directly. The LSVM technique performs classification into two classes. However, to perform a high dimensional classification for many classes, the multi-category classification approach using a kernel trick is required.

The Linear kernel trick transforms linearly non-separable data points into higher dimensional feature space where the data points cannot be linearly separated. Kernels are major strength to all SVM systems with small to moderate number of data. Linear kernel is the least common choice, though a powerful feature classifier machine that can be used to design linear functions based on the task at hand.

In this study, we designed a data-driven framework for reliable link classification that models KLSVM to produce stable and consistent results. We present the model based on the real dataset.

Our approach modeled and captured the two phenomena—data correlation and high dimensional data classification. Thus, we detected a faulty link and captured four outliers with extremely inconsistent behavior within the network. The technique has a good performance on link quality estimation accuracy.

Author Contributions: For this research article: Writing—Original Draft Preparation was done by L.M.M.; Writing—Review & Editing also was done by L.M.M.; under the supervision of Z.Z. and G.M.

Funding: This research received no external funding.

Acknowledgments: We could like to thank all the referees for their constructive comments and helpful suggestions. In addition, special thanks to M.Y. and Z.Z. for their valuable inputs on TinyOS and Matlab, respectively. Thank you J.P.A. for your valuable inputs on SVM. Finally, thank you Dr. M.W. for proofreading this paper.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SVM	Support Vector Machines
LSVM	Linear SVM
KLSVM	Kernelized Linear Support Vector Machine
SVC	Support Vector Classifier
WSN	Wireless Sensor Network
MEMS	Micro Electromechanical Systems
PRR	Packet Reception Rate
BCM	Beacon-based Measurement
NGMI	New Generation Mobile Internet Research Laboratory
CPDF	Conditional Probability Delivery Function
EMD	Earth Mover's Distance

References

1. Dargie, W.; Poellabauer, C. Motivation for a Network of Wireless Sensor Nodes. In *Fundamentals of Wireless Sensor Networks*; Wiley-Blackwell: Hoboken, NJ, USA, 2011; Chapter 1, pp. 1–16. [\[CrossRef\]](#)
2. Zidi, S.; Moulahi, T.; Alaya, B. Fault Detection in Wireless Sensor Networks Through SVM Classifier. *IEEE Sens. J.* **2018**, *18*, 340–347. [\[CrossRef\]](#)
3. Zhao, Z.; Kuendig, S.; Carrera, J.; Carron, B.; Braun, T.; Rolim, J. Indoor Location for Smart Environments with Wireless Sensor and Actuator Networks. In Proceedings of the 2017 IEEE 42nd Conference on Local Computer Networks (LCN), Singapore, 9–12 October 2017.
4. Shao, G.; Chen, J. A Load Balancing Strategy Based on Data Correlation in Cloud Computing. In Proceedings of the 9th International Conference on Utility and Cloud Computing (UCC'16), Shanghai, China, 6–9 December 2016; pp. 364–368. [\[CrossRef\]](#)
5. Yang, Y.; Wang, C. A novel method of data correlation analysis of the big data based on network clustering algorithm. In Proceedings of the 2015 IEEE International Conference on Communication Software and Networks (ICCSN), Chengdu, China, 6–7 June 2015; pp. 360–366. [\[CrossRef\]](#)
6. Han, X.; Du, Q. Interaction Between Big Data and Cognitive Science. In Proceedings of the 2nd International Conference on Compute and Data Analysis (ICCD 2018), DeKalb, IL, USA, 23–25 March 2018; pp. 1–5. [\[CrossRef\]](#)
7. Karthik, N.; Ananthanarayana, V.S. Data trust model for event detection in wireless sensor networks using data correlation techniques. In Proceedings of the 2017 Fourth International Conference on Signal Processing, Communication and Networking (ICSCN), Chennai, India, 16–18 March 2017; pp. 1–5. [\[CrossRef\]](#)
8. Zhang, Y.; Cheng, H.; Chen, D. Data Reconstruction with Spatial and Temporal Correlation in Wireless Sensor Networks. In Proceedings of the 3rd ACM Workshop on Mobile Sensing, Computing and Communication (MSCC'16), Paderborn, Germany, 5–8 July 2016; ACM: New York, NY, USA, 2016; pp. 40–51. [\[CrossRef\]](#)
9. Kim, S.M.; Wang, S.; He, T.; Kim, S.M.; Wang, S.; He, T. Exploiting Spatiotemporal Correlation for Wireless Networks Under Interference. *IEEE/ACM Trans. Netw.* **2017**, *25*, 3132–3145. [\[CrossRef\]](#)

10. Crary, N.; Tang, B.; Taase, S. Data Preservation in Data-Intensive Sensor Networks with Spatial Correlation. In Proceedings of the 2015 Workshop on Mobile Big Data (Mobidata'15), Hangzhou, China, 22–25 June 2015; ACM: New York, NY, USA, 2015; pp. 7–12. [\[CrossRef\]](#)
11. Di, M.; Joo, E.M. A survey of machine learning in Wireless Sensor networks from networking and application perspectives. In Proceedings of the 2007 6th International Conference on Information, Communications Signal Processing, Singapore, 10–13 December 2007; pp. 1–5. [\[CrossRef\]](#)
12. Rojo-Alvarez, J.L.; Martinez-Ramon, M.; Munoz-Mari, J.; Camps-Valls, G. Support Vector Machine and Kernel Classification Algorithms. In *Digital Signal Processing with Kernel Methods*; Wiley-Blackwell: Hoboken, NJ, USA, 2018; Chapter 10, pp. 433–502. [\[CrossRef\]](#)
13. Shawe-Taylor, J.; Sun, S. Chapter 16—Kernel Methods and Support Vector Machines. In *Academic Press Library in Signal Processing: Volume 1*; Diniz, P.S., Suykens, J.A., Chellappa, R., Theodoridis, S., Eds.; Elsevier: New York, NY, USA, 2014; Volume 1, pp. 857–881. [\[CrossRef\]](#)
14. Zoppis, I.; Mauri, G.; Dondi, R. Kernel Methods: Support Vector Machines. In *Reference Module in Life Sciences*; Elsevier: New York, NY, USA, 2018. [\[CrossRef\]](#)
15. Zhao, Z.; Dong, W.; Bu, J.; Gu, Y.; Chen, C. Link-Correlation-Aware Data Dissemination in Wireless Sensor Networks. *IEEE Trans. Ind. Electron.* **2015**, *62*, 5747–5757. [\[CrossRef\]](#)
16. Zhao, Z.; Xu, X.; Dong, W.; Bu, J. An Accurate Link Correlation Estimator for Improving Wireless Protocol Performance. *Sensors* **2015**, *15*, 4273–4290. [\[CrossRef\]](#) [\[PubMed\]](#)
17. Shu, J.; Liu, S.; Liu, L.; Zhan, L.; Hu, G. Research on Link Quality Estimation Mechanism for Wireless Sensor Networks Based on Support Vector Machine. *Chin. J. Electron.* **2017**, *26*, 377–384. [\[CrossRef\]](#)
18. Reeves, D.M.; Jacyna, G.M. Support vector machine regularization. *Wiley Interdiscip. Rev. Comput. Stat.* **2011**, *3*, 204–215. [\[CrossRef\]](#)
19. Shanthamallu, U.S.; Spanias, A.; Tepedelenioglu, C.; Stanley, M. A brief survey of machine learning methods and their sensor and IoT applications. In Proceedings of the 2017 8th International Conference on Information, Intelligence, Systems Applications (IISA), Larnaca, Cyprus, 27–30 August 2017; pp. 1–8. [\[CrossRef\]](#)
20. Ahmadi, H.; Bouallegue, R. Exploiting machine learning strategies and RSSI for localization in wireless sensor networks: A survey. In Proceedings of the 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC), Valencia, Spain, 26–30 June 2017; pp. 1150–1154. [\[CrossRef\]](#)
21. Chen, D.; Yuan, Z.; Hua, G.; Zheng, N.; Wang, J. Similarity Learning on an Explicit Polynomial Kernel Feature Map for Person Re-Identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.
22. Santos, I.; Devesa, J.; Brezo, F.; Nieves, J.; Bringas, P.G. OPEM: A Static-Dynamic Approach for Machine-Learning-Based Malware Detection. In Proceedings of the International Joint Conference CISIS'12-ICEUTE-12-SOCO12 Special Sessions, Ostrava, Czech Republic, 5–7 September 2012; Herrero, Á., Snášel, V., Abraham, A., Zelinka, I., Baruque, B., Quintián, H., Calvo, J.L., Sedano, J., Corchado, E., Eds.; Springer: Berlin/Heidelberg, Germany, 2013; pp. 271–280.
23. Rebentrost, P.; Mohseni, M.; Lloyd, S. Quantum Support Vector Machine for Big Data Classification. *Phys. Rev. Lett.* **2014**, *113*, 130503. [\[CrossRef\]](#) [\[PubMed\]](#)
24. Chen, J.; Deng, Y.; Bai, G.; Su, G. Face Image Quality Assessment Based on Learning to Rank. *IEEE Signal Process. Lett.* **2015**, *22*, 90–94. [\[CrossRef\]](#)
25. Saripalle, S.K.; Paiva, G.C.; Cliett, T.C.; Derakhshani, R.R.; King, G.W.; Lovelace, C.T. Classification of body movements based on posturographic data. *Hum. Mov. Sci.* **2014**, *33*, 238–250. [\[CrossRef\]](#) [\[PubMed\]](#)
26. Punitha, A.; Geetha, M.K. Texture based Emotion Recognition from Facial Expressions using Support Vector Machine. *Int. J. Comput. Appl.* **2013**, *80*. [\[CrossRef\]](#)
27. Lahmiri, S.; Boukadoum, M. New approach for automatic classification of Alzheimer's disease, mild cognitive impairment and healthy brain magnetic resonance images. *Healthc. Technol. Lett.* **2014**, *1*, 32–36. [\[CrossRef\]](#) [\[PubMed\]](#)
28. Ravale, U.; Marathe, N.; Padiya, P. Feature Selection Based Hybrid Anomaly Intrusion Detection System Using K Means and RBF Kernel Function. *Procedia Comput. Sci.* **2015**, *45*, 428–435. [\[CrossRef\]](#)
29. Khan, Z.U.; Hayat, M.; Khan, M.A. Discrimination of acidic and alkaline enzyme using Chou's pseudo amino acid composition in conjunction with probabilistic neural network model. *J. Theor. Biol.* **2015**, *365*, 197–203. [\[CrossRef\]](#) [\[PubMed\]](#)

30. Majumder, A.; Behera, L.; Subramanian, V.K. Emotion recognition from geometric facial features using self-organizing map. *Pattern Recognit.* **2014**, *47*, 1282–1293. [[CrossRef](#)]
31. Patel, J.; Shah, S.; Thakkar, P.; Kotecha, K. Predicting stock market index using fusion of machine learning techniques. *Expert Syst. Appl.* **2015**, *42*, 2162–2172. [[CrossRef](#)]
32. Hu, J.; Hu, J.; Lin, H.; Li, X.; Jiang, C.; Qiu, X.; Li, W. State-of-charge estimation for battery management system using optimized support vector machine for regression. *J. Power Sources* **2014**, *269*, 682–693. [[CrossRef](#)]
33. Yuan, X.; Chen, C.; Yuan, Y.; Huang, Y.; Tan, Q. Short-term wind power prediction based on LSSVM–GSA model. *Energy Convers. Manag.* **2015**, *101*, 393–401. [[CrossRef](#)]
34. Lv, Y.; Duan, Y.; Kang, W.; Li, Z.; Wang, F. Traffic Flow Prediction With Big Data: A Deep Learning Approach. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 865–873. [[CrossRef](#)]
35. Baghaee, H.R.; Mirsalim, M.; Gharehpetan, G.B.; Talebi, H.A. Nonlinear Load Sharing and Voltage Compensation of Microgrids Based on Harmonic Power-Flow Calculations Using Radial Basis Function Neural Networks. *IEEE Syst. J.* **2018**, 1–11. [[CrossRef](#)]
36. Chen, W.; Pourghasemi, H.R.; Naghibi, S.A. A comparative study of landslide susceptibility maps produced using support vector machine with different kernel functions and entropy data mining models in China. *Bull. Eng. Geol. Environ.* **2018**, *77*, 647–664. [[CrossRef](#)]
37. Hassan, U.K.; Nawwi, N.M.; Kasim, S. Classify a Protein Domain Using Sigmoid Support Vector Machine. In Proceedings of the 2014 International Conference on Information Science and Applications (ICISA), Seoul, South Korea, 6–9 May 2014; pp. 1–4. [[CrossRef](#)]
38. Yoo, H.; Shon, T. Novel Approach for Detecting Network Anomalies for Substation Automation based on IEC 61850. *Multimed. Tools Appl.* **2015**, *74*, 303–318. [[CrossRef](#)]
39. Cao, J.; Chen, T.; Fan, J. Fast online learning algorithm for landmark recognition based on BoW framework. In Proceedings of the 2014 9th IEEE Conference on Industrial Electronics and Applications, Hangzhou, China, 9–11 June 2014; pp. 1163–1168. [[CrossRef](#)]
40. Kalyan Kumar, V.P.; Suja, P.; Tripathi, S. Emotion Recognition from Facial Expressions for 4D Videos Using Geometric Approach. In *Advances in Signal Processing and Intelligent Recognition Systems*; Thampi, S.M., Bandyopadhyay, S., Krishnan, S., Li, K.C., Mosin, S., Ma, M., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 3–14.
41. Zhu, F.; Wei, J. Localization Algorithm in Wireless Sensor Networks Based on Improved Support Vector Machine. *J. Nanoelectron. Optoelectron.* **2017**, *12*, 452–459. [[CrossRef](#)]
42. Gui, G.; Pan, H.; Lin, Z.; Li, Y.; Yuan, Z. Data-driven support vector machine with optimization techniques for structural health monitoring and damage detection. *KSCE J. Civ. Eng.* **2017**, *21*, 523–534. [[CrossRef](#)]
43. Khan, F.; Memon, S.; Jokhio, S.H. Support vector machine based energy aware routing in wireless sensor networks. In Proceedings of the 2016 2nd International Conference on Robotics and Artificial Intelligence (ICRAI), Rawalpindi, Pakistan, 1–2 November 2016; pp. 1–4, doi:10.1109/ICRAI.2016.7791218. [[CrossRef](#)]
44. Ghaddar, B.; Naoum-Sawaya, J. High dimensional data classification and feature selection using support vector machines. *Eur. J. Oper. Res.* **2018**, *265*, 993–1004. [[CrossRef](#)]
45. Shao, Y.H.; Chen, W.J.; Zhang, J.J.; Wang, Z.; Deng, N.Y. An efficient weighted Lagrangian twin support vector machine for imbalanced data classification. *Pattern Recognit.* **2014**, *47*, 3158–3167. [[CrossRef](#)]
46. Ayadi, A.; Ghorbel, O.; Obeid, A.M.; Abid, M. Outlier detection approaches for wireless sensor networks: A survey. *Comput. Netw.* **2017**, *129*, 319–333. [[CrossRef](#)]
47. Gholipour, M.; Haghighat, A.T.; Meybodi, M.R. Hop-by-Hop Congestion Avoidance in wireless sensor networks based on genetic support vector machine. *Neurocomputing* **2017**, *223*, 63–76. [[CrossRef](#)]
48. Jie, C.; Zhiyi, F.; Guannan, Q.; Hongyu, S.; Dan, Z. An accurate traffic classification model based on support vector machines. *Int. J. Netw. Manag.* **2017**, *27*, e1962. [[CrossRef](#)]
49. Trinh, V.V.; Tran, K.P.; Huong, T.T. Data driven hyperparameter optimization of one-class support vector machines for anomaly detection in wireless sensor networks. In Proceedings of the 2017 International Conference on Advanced Technologies for Communications (ATC), Quy Nhon, Vietnam, 18–20 October 2017; pp. 6–10. [[CrossRef](#)]
50. Salberg, A. Detection of seals in remote sensing images using features extracted from deep convolutional neural networks. In Proceedings of the 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Milan, Italy, 26–31 July 2015; pp. 1893–1896. [[CrossRef](#)]

51. Huang, G.; Zhang, B.; Yao, Z. Data correlation aware opportunistic routing protocol for wireless sensor networks. In Proceedings of the 2017 IEEE International Conference on Communications (ICC), Paris, France, 21–25 May 2017; pp. 1–6. [\[CrossRef\]](#)
52. Huang, Z.; Xie, T.; Zhu, T.; Wang, J.; Zhang, Q. Application-driven sensing data reconstruction and selection based on correlation mining and dynamic feedback. In Proceedings of the 2016 IEEE International Conference on Big Data (Big Data), Washington, DC, USA, 5–8 December 2016; pp. 1322–1327. [\[CrossRef\]](#)
53. Kang, X.; Liu, L.; Ma, H. Data correlation based crowdsensing enhancement for environment monitoring. In Proceedings of the 2016 IEEE International Conference on Communications (ICC), Kuala Lumpur, Malaysia, 22–27 May 2016; pp. 1–6. [\[CrossRef\]](#)
54. Kumar, S.; Kumar, S. Data aggregation using spatial and temporal data correlation. In Proceedings of the 2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE), Noida, India, 25–27 February 2015; pp. 479–483. [\[CrossRef\]](#)
55. Kim, S.M.; Wang, S.; He, T. cETX: Incorporating Spatiotemporal Correlation for Better Wireless Networking. In Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems (SenSys'15), Seoul, Korea, 1–4 November 2015; ACM: New York, NY, USA, 2015; pp. 323–336. [\[CrossRef\]](#)
56. Erfani, S.M.; Rajasegarar, S.; Karunasekera, S.; Leckie, C. High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning. *Pattern Recognit.* **2016**, *58*, 121–134. [\[CrossRef\]](#)
57. Jedlinski, L.; Jonak, J. Early fault detection in gearboxes based on support vector machines and multilayer perceptron with a continuous wavelet transform. *Appl. Soft Comput.* **2015**, *30*, 636–641. [\[CrossRef\]](#)
58. Deng, F.; Guo, S.; Zhou, R.; Chen, J. Sensor Multifault Diagnosis With Improved Support Vector Machines. *IEEE Trans. Autom. Sci. Eng.* **2017**, *14*, 1053–1063. [\[CrossRef\]](#)
59. Garcia-Font, V.; Garrigues, C.; Rifà-Pous, H. A Comparative Study of Anomaly Detection Techniques for Smart City Wireless Sensor Networks. *Sensors* **2016**, *16*, 868. [\[CrossRef\]](#) [\[PubMed\]](#)
60. Wang, S.; Lu, S.; Dong, Z.; Yang, J.; Yang, M.; Zhang, Y. Dual-Tree Complex Wavelet Transform and Twin Support Vector Machine for Pathological Brain Detection. *Appl. Sci.* **2016**, *6*, 169. [\[CrossRef\]](#)
61. Abu-Mostafa, Y.S.; Magdon-Ismael, M.; Lin, H.T. Support Vector Machines. In *Learning from Data*; Wiley-Blackwell: Hoboken, NJ, USA, 2006; Chapter 9, pp. 404–466. [\[CrossRef\]](#)
62. Rubner, Y.; Tomasi, C.; Guibas, L.J. A metric for distributions with applications to image databases. In Proceedings of the Sixth International Conference on Computer Vision, Bombay, India, 4–7 January 1998; pp. 59–66. [\[CrossRef\]](#)
63. Raphael, S., Causality and Correlation. In *The Wiley-Blackwell Encyclopedia of Social Theory*; American Cancer Society: Atlanta, GA, USA, 2017; pp. 1–4. [\[CrossRef\]](#)

© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).