Adaptive Spike Detection for Resilient Data Stream Mining

Clifton Phua[1] **Kate Smith-Miles**[2] **Vincent Lee**[1] **Ross Gayler**[3]

[1] Clayton School of Information Technology
Monash University,
Clayton, Victoria, Australia 3800,
Email: `clifton.phua@infotech.monash.edu.au, vincent.lee@infotech.monash.edu.au`

[2] School of Engineering and Information Technology
Deakin University,
Burwood, Victoria, Australia 3125,
Email: `katesm@deakin.edu`

[3] Veda Advantage
Level 12, 628 Bourke Street
Melbourne, Victoria, Australia 3000,
Email: `ross.gayler@vedaadvantage.com`

> "A system's resilience is the single most important security property it has."
>
> - Bruce Schneier, 2003, "Beyond Fear: Thinking Sensibly about Security in an Uncertain World"

Abstract

Automated adversarial detection systems can fail when under attack by adversaries. As part of a resilient data stream mining system to reduce the possibility of such failure, adaptive spike detection is attribute ranking and selection without class-labels. The first part of adaptive spike detection requires weighing all attributes for spiky-ness to rank them. The second part involves filtering some attributes with extreme weights to choose the best ones for computing each example's suspicion score. Within an identity crime detection domain, adaptive spike detection is validated on a few million real credit applications with adversarial activity. The results are F-measure curves on eleven experiments and relative weights discussion on the best experiment. The results reinforce adaptive spike detection's effectiveness for class-label-free attribute ranking and selection.

Keywords: adaptive spike detection, resilient data mining, data stream mining, class-label-free attributes ranking and selection

1 Introduction

Adversarial detection systems are fraud and crime detection, and other security systems. Our main concern here is when adversaries focus their attack on certain attributes (also known as fields, variables, and features), the weights (importance) of attributes can change quickly.

Data stream mining (Kleinberg 2005) involves detecting real-time patterns to produce accurate suspicion scores (which are indicative of anomalies). At the same time, the detection system has to handle continuous and rapid examples (also known as records, tuples, and instances) where the recent examples have no class-labels.

The work here is motivated by identity crime detection, or more specifically, credit application fraud detection (Phua et al. 2005) (also known as white-collar crime) . When adversaries manipulate real-time data, these detection systems can fail badly, if not completely. First, this is caused by too many new and successful attacks which are detected too late. Second, this is due to time delays in manual intervention by trusted people when new attacks are detected and underway. Resilient data stream mining is necessary to prevent failure of detection systems. It is the security systems' ability to degrade gracefully, or to adjust to changing circumstances when under attack (Schneier 2003).

Resilient data stream mining requires a series of multiple, independent, and sequential layers in a system. This is termed "defence-in-depth". These layers interact with each other to deal with the new and deliberate attacks, and make it much harder for persistent adversaries to circumvent the security system (Schneier 2003).

For example, there is a need to protect the personal identity databases of financial institutions. They contain individual applicants' details from real identity theft and synthetic identity fraud. The former refers to innocent peoples' identity details being used illegitimately by adversaries without their permission. The latter refers to non-existent peoples' identity details being created by adversaries to cheat assessment procedures. Three of our proposed identity crime detection procedures are:

- Known fraud matching (Phua et al. 2005) as **first-layer defence** - it is effective for repetitive frauds and real identity theft. However, there is a long delay between time that the identity is stolen and time the identity is actually reported stolen. This allows adversaries to use any stolen identity quickly before being discovered.
- Communal detection (Phua et al. 2006b) as **second-layer defence** - it utilises an example-based approach (similar to graph theory and record linkage) by working on a fixed set of attributes. It reduces the significant time delay and false alarms by filtering normal human relationships with whitelists to save significant money. In addition, it is good for new, duplicative frauds and synthetic identity fraud. But communal de-

Copyright ©2007, Australian Computer Society, Inc. This paper appeared at the Sixth Australasian Data Mining Conference (AusDM 2007), Gold Coast, Australia. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 70, Peter Christen, Paul Kennedy, Jiuyong Li, Inna Kolyshkina and Graham Williams, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

tection is domain-specific, inflexible, and computationally slow.

- Spike detection (Phua et al. 2006a) as **third-layer defence** - it uses an attribute-oriented approach (similar to time series analysis) by working on a variable-size set of attributes. It reduces significant time delay by searching for recent duplicates. In comparison to communal detection without blocking (Baxter et al. 2003), spike detection with string similarity matching is computationally faster.

There is a fundamental security flaw with our communal detection framework (Phua et al. 2006b). It captures a substantial amount of frauds by filtering innocent relationships and data errors. However, if three or more values of each current identity are exact or similar to a window of previously collected identities, then a numeric suspicion score is produced. However, this encourages adversaries to quickly duplicate one and/or two important values which have been successful in their previous attempts. Spike detection can overcome this weakness by monitoring on one or two of the most important attributes.

Spike detection, with exact matching on a few of the most important attributes is much faster than communal detection. In hindsight, these important attributes have few missing values, are not de-identified, and string similarity matching can be performed (anonymised from identity strings to unidentifiable numbers). These attributes are appropriate ("not-too-dense and not-too-sparse") so that once they become dense, they become suspicious. They are also easiest to investigate (contacting the person or verifying against other databases). In contrast, communal detection scans many attributes of each identity as they arrive continuously and rapidly, so the system can become too slow with increased volume.

Both spike and communal detection have their own unique advantages. Spike detection can be computed in parallel on multiple workstations, each on an attribute. Also, the spike detection's relative weights can be applied to the communal detection's attributes. Communal detection is more accurate as it filters real communal relationships and data errors from other more anomalous exact and approximate duplicates. In contrast, spike detection can only remove duplicates based on each attribute.

At the highest level, resilient data stream mining also protects each security layer individually in the system. This is introduced in this paper as "layer reinforcement". This is to reduce the effects of attacks on security layers by persistent adversaries. For example, each of our identity crime detection procedure is defended by:

- Personal name analysis (Phua et al. 2006) as the **second-layer reinforcement** - it focuses on verifying and extracting information from personal names to improve known fraud matching.

- Adaptive communal detection (Phua et al. 2007) as the **third-layer reinforcement** - it has been proposed to prevent tampering of whitelists in communal detection.

- Adaptive spike detection is proposed in this paper as the **fourth-layer reinforcement** - static spike detection will be probed by persistent adversaries and the exact attributes monitored will eventually be known and circumvented with more duplicated values of the other attributes instead. To dynamically adapt to these adversaries' counter-measures, this paper formulates two related and consecutive challenges in adaptive spike detection.

The first challenge is to rank all attributes with class-labels. Although classification algorithms provide accurate attribute ranking with the benefit and clarity of hindsight, three main problems exist with using class-label attribute ranking for security detection systems:

- **Untimely:** There are time delays in labeling examples as positive because it takes time for fraud/crime to be revealed. This provides a window of opportunity for adversaries. Class-label attribute ranking is also computationally slow in an operational event-driven environment which requires efficient processing and rapid decision making (Phua et al. 2005).

- **Incomplete:** The positive class can be significantly understated. The case where the system labels current and prior positives but not future ones is common (Phua et al. 2006c). It is also possible that some of the data sources do not contribute positive class-labels.

- **Un-reliable:** The class-labeling is highly dependent on people. Each example has to be manually labeled and this has the potential of breaching privacy particularly if the data contains identity information. In addition, human factors can result in class-labels being incorrect, expensive, and difficult to obtain (Phua et al. 2005).

The second challenge is to use some attributes to detect attacks. The argument here is that two extreme (can also be known as redundant) types of attributes do not provide any symptoms of attacks:

- **Densest attributes:** They are attributes with highest weights (also known as densest; most spiky, duplicative, repetitive, and highest regularity and frequency attributes). They are usually numerical and have a finite number of values and therefore occur more frequently (for example, street numbers and postcodes).

  Since all their values are already highly duplicative, the system cannot find significantly denser values which are highly indicative of suspicious activity.

- **Sparsest attributes:** They are attributes with the smallest weights. They usually consist of string occurrences and identifiers with an enormous number of possible values which can occur at widely spaced intervals (for example, personal names and telephone numbers).

  Since their values do not re-occur or occur so rarely, the system cannot detect many attacks.

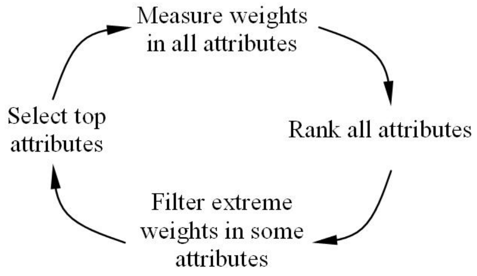
Measure weights in all attributes
Rank all attributes
Filter extreme weights in some attributes
Select top attributes

Figure 1: Attribute selection cycle

Figure 1 gives a visual account of the general iterative steps to ensure good class-label-free attribute ranking and selection. There are two research questions for adaptive spike detection for security systems:

Question 1 - How does the system empirically measure the weights of all attributes and rank them without the class-label attribute?

Question 2 - How does the system systematically filter redundant weights and select some appropriate attributes to calculate the suspicion score?

Section 2 outlines related work. Section 3 introduces the re-designed spike detection framework; and explains its resilient version in Section 4. Section 5 describes the identity crime detection domain, electronic credit application data, experimental design, and the performance metric; followed by results and discussion in Section 6. Section 7 concludes the paper.

2 Related Work

Spike detection is inspired by Stanford Stream Data Manager (STREAM) (Balakrishnan et al. 2004) and AURORA (Arasu et al. 2003) which deal with a rapid and continuous stream with structured examples by executing continuous queries. STREAM uses the SQL extension - Continuous Query Language (CQL) - in the extraction of example-based (or time-based) sliding windows, optional elimination of exact duplicates, and enforcement of increasing time-stamp order in queues. AURORA has been applied to financial services, highway toll billing, battalion and environmental monitoring.

Analysing sparse attributes exists in document and bio-surveillance streams. Kleinberg (2005, 2002) surveys threshold-, state-, and trend-based stream mining techniques used in topic detection and tracking. Goldenberg et al. (2002) use time series analysis to track early symptoms of synthetic anthrax outbreaks from daily sales of retail medication (throat, cough, and nasal) and some grocery items (facial tissues, orange juice, and soup).

Adaptive spike detection is easily extensible. In addition to identity crime detection, they are useful to other well-known security domains which profile suspicious behaviour or activity. These domains also aim to detect early irregularities in temporal data with unsupervised techniques. They include, but not limited to:

- **Bio-terrorism (also known as syndromic surveillance, aberration, and outbreak) detection:** Control-chart-based statistics, exponential weighted moving averages, and generalised linear models were tested on the same benchmark data and fixed alert rate. The conclusion was that these techniques perform well only when there are rapid and large increases in duplicates relative to the baseline level (Jackson et al. 2007). Bayesian networks were applied to uncover simulated anthrax attacks from real emergency department data (Wong et al. 2003).

- **Credit transactional account fraud detection:** Peer Group Analysis is recommended to monitor inter-account behaviour over time and also suggest Break Point Analysis to monitor intra-account behaviour over time (Bolton & Hand 2001).

- **Spam detection:** The use of document space density (class-labels are avoided) to find large volumes of similar emails is encoded as hash-based text (Yoshida et al. 2004).

3 Spike Detection Framework

The spike detection framework (Phua et al. 2006a) monitors streams to find recent and denser values in attributes (which can be highly indicative of identity crime). In other words, spiky-ness of each attribute is more than the number of exact and approximate values - it also factors in the recency of the duplicates. The more current the duplicates, the more interesting or suspicious they become. The framework basically uses exponential smoothing to find single attribute value spikes, and integrates the multiple attribute value spikes to score each example.

Input	Process	Output
Current example d_n Window W Window steps k Exponential smoothing α Similarity threshold r Time filter e	Process each current value against previous values within an example-based window Step 1: Calculate scaled counts of current value by comparison to previous examples window[1] Step 2: Calculate smoothed score on current value[2] Step 3: Calculate suspicion score on current example[3]	Score

Table 1: Parameters, spike detection, and suspicion score

Table 1 gives an overview of the input parameters, spike detection process/algorithm, and output suspicion score.

3.1 Step 1: Scaled Counts in Single Step

Let Y represent one continuous stream with an ordered set of $\{\ldots, y_{j-2}, y_{j-1}, y_j, y_{j+1}, y_{j+2}, \ldots\}$ discrete streams. Let y_j represent a current discrete data stream with an ordered set of $\{d_{j,n}, d_{j,n+1}, d_{j,n+2}, \ldots, d_{j,n+p}\}$ examples to be processed in real time. For simplicity, the subscript j is omitted. Each current example d_n (to be scored) contains M chosen attributes with a set of $\{a_{n,1}, a_{n,2}, \ldots, a_{n,M}\}$ values. Let W represent the window size of number of previous examples to match against.

Let r represent the string similarity (also known as fuzzy matching) threshold between values where $0 < r \leq 1$. 0 is a complete mismatch and 1 is an exact match. The fast string similarity metric Jaro-Winkler (Winkler 2006) is used. Let e represent the time difference filter (for example, seconds, minutes, and hours) between values to improve data quality where $0 \leq e << \leq$ inf. 0 means no filter and inf means all previous values are filtered.

In addition to a larger number of attributes M or window size W, lower string similarity threshold r or time difference filter e might also produce higher suspicion score for an example $S(d_n)$. Each current value a_{n-1} is being searched for its exact or approximate values in $\{a_{n-1,i}, a_{n-2,i}, \ldots, a_{n-W,i}\}$ where the matches' string similarity and time difference are larger than r and e.

$$s_x(a_{n,i}) = count_x(a_{n,i})/k, \text{ for } x = 1, 2, \ldots, t \quad (1)$$

In reference to Phua et al. (2006a): [1]At step 1, no form of random sampling is used except to filter out six dummy and three hashed-addresses subscribers. [2]At step 2, "smoothing level", "spiking alpha", and four other optional parameters are removed to simplify work. [3]At step 3, explicit normalisation of scores to 0 and 1 is not necessary.

Equation 1 is the scaled counts for each step $s_x(a_{n,i})$ (a window is made up of many steps) to remove volume effects where $0 \leq s_x(a_{n,i}) \leq 1$. Each W is divided into t steps (number of blocks of consecutive values) of k step size (maximum number of values in each block). t is also the most recent time step.

3.2 Step 2: Spike Detection of Single Value

$$S(a_{n,i}) = \sum_{x=1}^{t} [(1-\alpha) \times s_x(a_{n,i}) + \alpha \times s_{x-1}(a_{n,i})] \quad (2)$$

Equation 2 is the exponential smoothing of each value (all steps) to determine spike score $S(a_{n,i})$ for weighing current examples more heavily or previous examples more lightly (Cortes et al. 2003) where $0 \leq S(a_{n,i}) \leq 1$. α is exponential smoothing factor to gradually discount the effects of previous older steps and $0 \leq \alpha \leq 1$.

3.3 Step 3: Suspicion Score from Multiple Values

$$S(d_n) = \sum_{i=1}^{M} S(a_{n,i}) \quad (3)$$

Equation 3 sums up all the spike scores to derive a suspicion score for each example $S(d_n)$ where $0 \leq S(d_n) \leq M$.

3.4 A Simple Spike Detection Illustration

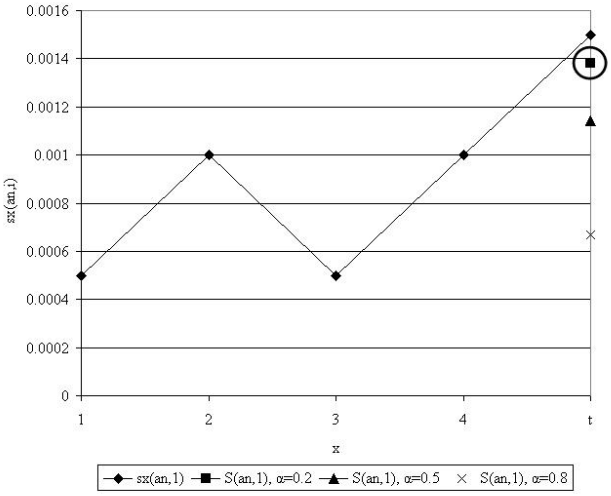

Figure 2: Scaled Counts and Spike Detection

Figure 2 demonstrates steps 1 and 2 of the spike detection framework. The y-axis represents the scaled counts; and the x-axis represents the steps. In the illustration above, given that the parameters' values are $W = 10{,}000, t = 5, k = W/t = 2{,}000$, therefore $count_{1,2,\ldots,5}(a_{n,1}) = 1, 2, 1, 2, 3$.

Step 1: Scaled Counts $count_{1,2,\ldots,5}(a_{n,1}) = 0.0005, 0.001, 0.0005, 0.001, 0.0015$ is represented by the line.

Step 2: Spike Detection $S(a_{n,1}) = 0.0013$ is the smoothed spike score (circled point) with the lowest weight on the previous examples ($\alpha = 0.2$).

Figure 3 shows step 3 of the spike detection framework. The y-axis represents the cumulative suspicion score; and the x-axis represents the number of attributes. For example, given that $S(a_{n,1,2,\ldots,5}) = 0.0013, 0.0129, 0.00543, 0.0511, 0.0732$.

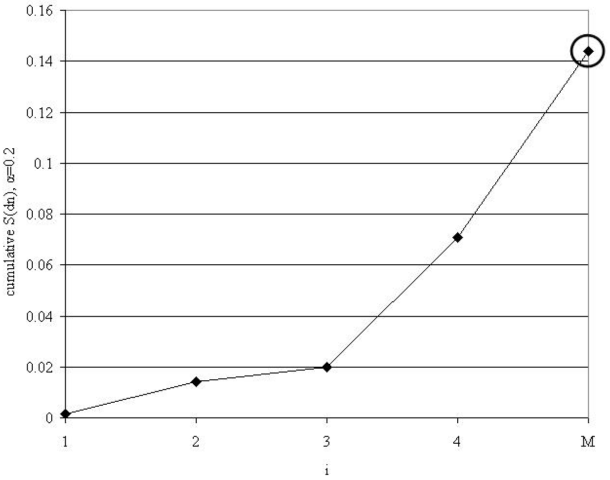

Figure 3: Suspicion score

Step 3: Suspicion Score $S(d_n) = 0.0013 + 0.0129 + 0.00543 + 0.0511 + 0.0732 = 0.144$ (circled point).

4 Adaptive Spike Detection

With adversarial activity in mind, the weight of each attribute is measured regularly at either fixed time or fixed example intervals (for example, after each month or after every ten thousand examples) to re-weigh and re-rank all attributes. To be more specific, each attribute is measured by relative weights at every interval.

From all attributes' weights, attributes with highest weights (densest) and lowest weights (sparsest) are filtered. Therefore, suspicion scores are computed from "not-too-dense and not-too-sparse" attributes with fewer missing values (so that the scores are more accurate).

In this way, when these appropriate attributes' weights suddenly become larger, it creates a spike in the time series, making them more interesting or suspicious. To be precise, attributes with relative weights which exceed one standard deviation or below half of average will have their weights set to zero. In this way, only some attributes are re-chosen and have their corresponding non-zero weights factored into the suspicion score.

4.1 Initialisation of Weights

$$\bar{w}_i = 1/M \quad (4)$$

When there are no prior weights, Equation 4 uses average/equal weights for all attributes.

4.2 Application of Weights

$$S(d_n) = \sum_{i=1}^{M} [\hat{w}_i \times S(a_{n,i})] \quad (5)$$

When processing each example, Equation 5 which is an extension from Equation 3, applies relative weights to all corresponding attribute values of each example.

4.3 Evolution of Weights

$$w_i = \sum_{n=1}^{p} S(a_{n,i})]/n \quad (6)$$

When updating weights at the end of each interval, Equation 6 represents the absolute/total weights per example.

$$\bar{w}_i = w_i / \sum_{i=1}^{M} w_i \qquad (7)$$

Modified from Equation 6, Equation 7 represents the relative weights.

$$\hat{w}_i = \begin{cases} \bar{w}_i & \text{if average weight}/2 \leq \bar{w}_i \\ & \leq \text{average weight} + \text{standard deviation} \\ 0 & \text{otherwise} \end{cases} \qquad (8)$$

Modified from Equation 7, Equation 8 symbolises the filtered relative weights which are actually applied to the attributes in Equation 5. Equation 8 retains only the relative weights of attributes which remain within the lowerbounds (average weight is usually low) and upperbounds (to exclude only the densest attributes), and removes attributes with extreme relative weights by setting zero weights.

4.4 A Simple Weights Illustration

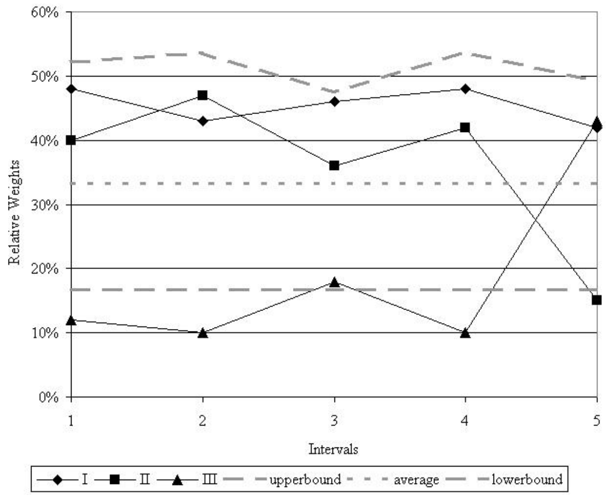

Figure 4: Evolution of relative weights

Figure 4 explains the concept of attributes' relative weights and ranks changing over time. The y-axis represents relative weights - the density/spiky-ness of three attributes - I II, and III - with respect to one another; and the x-axis represents the time interval (for example, hours, days, and months).

In the first four intervals, attributes I and II are ranked higher than III. As they are also within the upper and lower boundaries, they are chosen to calculate the suspicion score. However, in the last interval, attribute II loses its density at the expense of III and falls below the lower bound limit. As attribute III suddenly becomes significantly denser (deemed as anomalous and quite possibly a new attack), therefore it is used together with attribute I to calculate subsequent suspicion scores.

5 Identity Crime Detection

Adaptivity is about helping the system adjust and function well within a changing environment. A data streaming environment is a rapidly changing one and the weights (importance) of attributes do not remain static. Therefore, spike detection is necessary to measure an attribute value's regularity within its attribute's recent times and represent them as weights relative to all other weights.

In addition, adversaries gather information about previous parameters of the spike detection framework to choose attributes that attempt to force the worst-case result. Those attributes with the appropriate amount of regularity for detecting suspicious behaviour do change in a principled fashion. Knowing them in time are additional defences against adversaries.

Therefore, adaptive spike detection is a specific case of resilient data stream mining. It deals with our adversaries who have a tendency to re-use identity values of certain attributes in a bursty manner. In addition, it also copes well when adversaries change their focus to other attributes. In this way, the adversaries are more likely to get relatively higher suspicion scores which are directly correlated with the risk of identity crime (see Figure 6).

5.1 Data and Evaluation Metrics

There are five main technical challenges posed by data used in the following experiments:

- **Large scale:** Thirteen months of several million real credit applications (only the last seven months are used in the experiments described here because they have the most complete class labels). Every month is made up of a few hundred thousand applications and every day has more than ten thousand applications. The data is recent, consecutive, and time-stamped to the milliseconds.

- **Dense and sparse attributes:** About thirty raw attributes such as personal names, addresses, telephone numbers, driver licence numbers (or social security numbers), date-of-births, and other personal identifiers. Some of these personally identifying attributes were encrypted prior to this study to preserve privacy. Encrypted attributes can be exactly matched, but in a real application unencrypted attributes would be used to allow approximate matching. For confidentiality reasons, we cannot specify the best attributes found in this study for credit application fraud detection.

- **Extreme class imbalance:** Less than one percent of these are known to be fraudulent in binary class-labeled (as "fraud" or "legal") data. Also, the earliest and latest months' known fraud rate is significantly understated as not all known frauds were provided for this research.

- **Diverse data sources:** A few dozen financial institutions are providers of examples. Each provider has varying arrival rates, has sudden behavioural changes, and contributes their own number and type of attributes, and adds data quality problems.

- **Few significant fraud patterns:** For the period under analysis, relational (links between examples), temporal (for example, hourly, daily, and monthly), spatial (for example, suburb, country, and state), and provider-related fraud behaviour are hard to differentiate from legitimate behaviour.

For evaluation metrics, precision-recall curves are avoided as they will divulge the sensitive nature of the true positive tp, false positive fp, and false negative fn rates.

Also, metrics which use true negatives tn such as accuracy and receiver operating characteristic curves are avoided since fp rates are likely to be understated (Christen & Goiser 2007).

$$F\text{-measure curve} = \frac{2 \times precision_X \times recall_X}{precision_X + recall_X} \quad (9)$$

The F-measure curve in Equation 9 consists of multiple values under X different thresholds. Each value depicts a trade-off between $precision_X = \frac{tp}{tp+fp}$ and $recall_X = \frac{tp}{tp+fn}$.

5.2 Experimental Design

In the following spike detection experiments, we are particularly interested in finding out which are the best attributes, out of a total of 19, for detecting identity crime. To do so, the parameters, applied to each attribute, for all the following experiments remain unchanged:

- Window $W = w/100$, Window Steps $k = 10$
- Exponential Smoothing $\alpha = 0.5$
- Similarity Threshold $r = 0.8$, Time Filter $e = 1$ hour

Some of their values, such as w, $\alpha = 0.5$, and $r = 0.8$, are based on our previous experimental experience and current practical domain knowledge. However, in comparison with our previous experiments in communal and spike detection, parameter values of W and k here are much smaller - W is 100 times smaller, k is 10 times smaller - to compare fewer examples. String similarity matching can be performed on 10 attributes; but cannot be applied to the other 9 because they seem to be too dense and/or are deidentified. Also, e is larger to filter more examples.

The use of small W and k values are due to the very high computational cost in applying string similarity to all comparisons for the 10 attributes. In addition, the experiments are meant to illustrate the concepts of adaptive spike detection. Also, for confidentiality reasons, these experiments do not reveal the results of a realistic W and k.

t Exp.	Attribute(s)
t1	I
t2	IV
t3	V
t4	XIV
t5	XVII
t6	XVIII
t7	XIX
t8	All-static
t9	2-static (XIV & XVIII)
t10	All-monthly
t11	2-monthly

Table 2: Static (t1 to t9) and adaptive (t10 and t11) spike detection experiments to test predictability of attributes

Table 2 show that experiments t1 to t9 are static where the relative weights are not used. t1 to t7 uses individual attributes regarded as useful from domain knowledge. t8 uses all 19 attributes. t9 uses only top 2 attributes (XIV and XVIII as advised by domain experts) throughout all the data.

Experiments t10 and t11 are adaptive where the relative weights change at a monthly interval. t10 answers Question 1 by measuring weights and ranking all attributes monthly without the class-label attribute. t11 answers both Questions 1 and 2 by filtering extreme ranked weights and then choosing the top 2 attributes (either XII, XIV, XIX, or III according to the highest unfiltered relative weights) monthly to calculate the suspicion score.

6 Results and Discussion

With F-measure over different thresholds, results are presented from seven important attributes, and justifies importance of the adaptive spike detection framework to measure and rank attributes. With relative weights, the useful role of adaptive spike detection to filter and choose attributes are verified.

6.1 Spike Detection Results

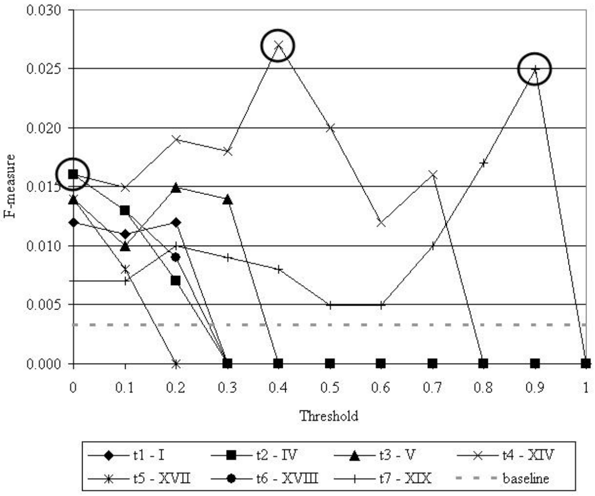

Figure 5: F-measure across 11 thresholds, of spike detection experiments t1 to t7 of individual attributes

Figure 5 illustrates the F-measure results over different thresholds of seven valuable single attributes. The most predictive attribute by spike detection is t4 - XIV with F-measure above 0.025 at threshold 0.4. This fact is also acknowledged by domain experts.

There is a need to automatically filter out most attributes for calculation of the suspicion score. Although all the other individual attributes are better than the baseline (random) at threshold 0, most are much poorer attributes compared to attribute XIV across most thresholds.

Spike detection is practical for our security domain. Two other predictive attributes revealed by spike detection include t7 - XIX and t6 - XIV. At the higher thresholds, the attribute XIX yield better results than all other attributes. Attribute XIX is another predictive attribute acknowledged by domain experts.

Figure 6 illustrates the F-measure results over big and small sets of attributes, static and adaptive. From observation, the most predictive set of attributes is t11 - 2-monthly with F-measure above 0.025 at threshold 1.

Finding the right set of appropriate attributes is crucial. This is illustrated by two facts from the F-measure results: First, the use of 2 attributes (static or adaptive) performs better than using all attributes. Second, 2-monthly is superior to using just the most predictive attribute XIV with the former's F-measure above 0.02 for most thresholds.

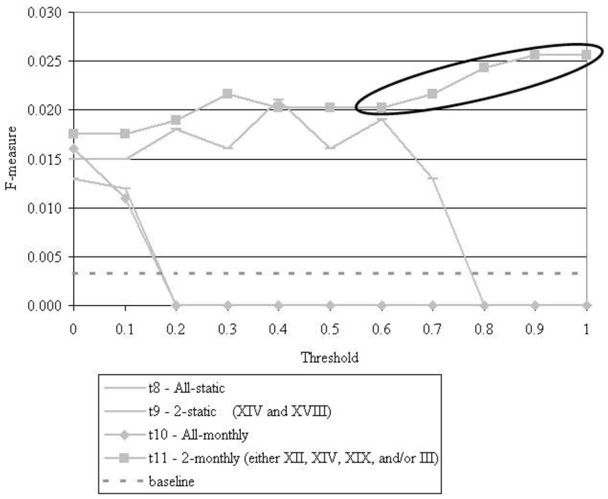

Figure 6: F-measure across 11 thresholds, of static (t8 and t9) and adaptive (t10 and t11) spike detection experiments of multiple attributes

Adaptivity with changing relative weights for attributes provide better results than static attributes with no weights. This is substantiated by t11 - 2-monthly which outperforms the second-best result from t9 - 2-static by a large margin.

In reality, F-measure results will be higher. The current F-measure results are underestimated because many of the class-labels were still not known at the time when this data set was constructed. Hence, the F-measure performance metric evaluated predictions based on significantly smaller numbers of positive class-labels (Phua et al. 2007).

6.2 Relative Weights Discussion

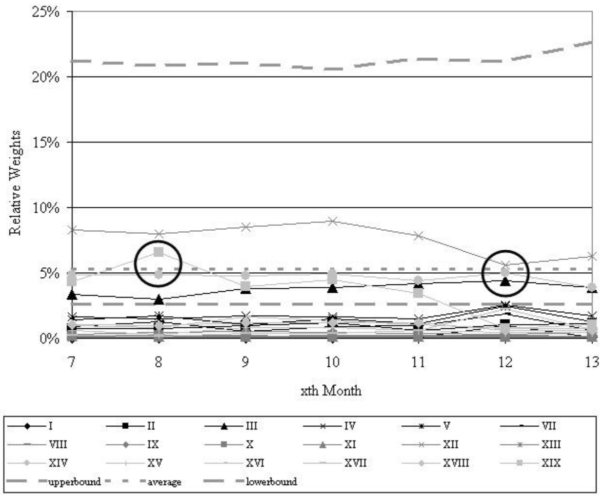

Figure 7: Relative weights across 7 months, of all attributes (except attribute VI) from experiment t11 - 2-monthly

Figure 7 highlights the relative weights which are within acceptable boundaries across seven months (the reason that a smaller set of attributes, surprisingly, performs better). Use of relative weights find the most appropriate attributes. Only a maximum of four attributes stay within the lower and upperbounds for any given month. The two best attributes XIV and XIX which are tested in Figure 5 have the highest acceptable relative weights.

Relative weights change over time and so do appropriate attributes: attribute XIX overtook XIV during the 8th month as the top 2 attributes and it dropped out of the acceptable boundaries at the 12th month.

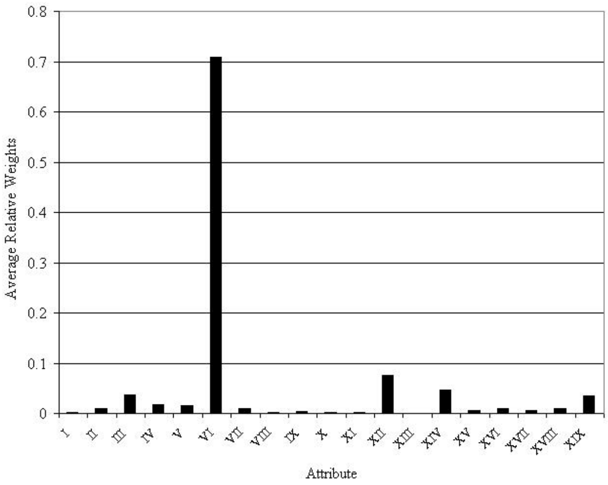

Figure 8: Average relative weights of all 7 months, of all attributes (inclusive of attribute VI) from experiment t11 - 2-monthly

Figure 8 focuses attention on the average relative weights of all seven months. The densest and sparsest attributes are not predictive. Attribute VI overspikes (highest weight). The other attributes do not spike enough (smallest weights). Yet, the densest and sparsest attributes cannot be discarded permanently because they can still become useful in the future.

There is a strong relationship between spiky-ness of certain attributes (represented by acceptable relative weights) and risk of identity crime (represented by class-labels). This is evident from the attributes XII, XIV, III, and XIX which are both spiky and predictive of fraud/crime. Therefore, the interpretation of results on substantial amount of historical data, modeled as data streams, justify that adaptive spike detection is significantly better than the static version (see Figure 6). As this idea is novel, attackers cannot apply what they have studied previously from elsewhere.

7 Conclusion

The overall goal is to propose resilient data stream mining for all data mining-based security systems with adaptive spike detection for attribute ranking and selection. The spike detection framework's parameters and suspicion score functions were significantly updated and made more resilient with the evolution of weights. In our identity crime domain, the challenges in our real data and the rationale for the evaluation measure were given. In the spike detection results, adaptive spike detection's attribute ranking and selection gave the best outcome. A deeper analysis of the relative weights showed that the most appropriate attributes were found.

8 Acknowledgements

The first author is financially supported by ARC under Linkage Grant Number LP0454077 and previously by DEST under an Endeavour Research Fellowship.

References

Arasu, A., Babcock, B., Babu, S., Datar, M., Ito, K., Nishizawa, I., Rosenstein, J. & Widom, J. (2003), STREAM: the stanford stream data manager demonstration description, *in* 'SIGMOD03'.

Balakrishnan, H., Balazinska, M., Carney, D., Cetintemel, U., Cherniack, M., Convey, C., Galvez, E., Salz, J., Stonebraker, M., Tatbul, N., Tibbetts, R. & Zdonik, S. (2004), 'Retrospective on aurora', *VLDB Journal* **13**(4), pp. 370–383.

Baxter, R., Christen, P. & Churches, T. (2003), A comparison of fast blocking methods for record linkage, *in* 'ACM SIGKDD03 Workshop on Data Cleaning, Record Linkage and Object Consolidation'.

Bolton, R. & Hand, D. (2001), Unsupervised profiling methods for fraud detection, *in* 'Credit Scoring and Credit Control VII'.

Christen, P. & Goiser, K. (2007), Quality and complexity measures for data linkage and deduplication, *in* F. Guillet & H. Hamilton, eds, 'Quality Measures in Data Mining', Springer.

Cortes, E., Pregibon, D. & Volinsky, C. (2003), 'Computational methods for dynamic graphs', *Journal of Computational and Graphical Statistics* **12**, pp. 950–970.

Goldenberg, A., Shmueli, G. & Caruana, R. (2002), 'Using grocery sales data for the detection of bioterrorist attacks', *Statistical Medicine*.

Jackson, M., Baer, A., Painter, I. & Duchin, J. (2007), 'A simulation study comparing aberration detection algorithms for syndromic surveillance', *BMC Medical Informatics and Decision Making* **7**(6).

Kleinberg, J. (2005), Temporal dynamics of on-line information streams, *in* M. Garofalakis, J. Gehrke & R. Rastogi, eds, 'Data Stream Management: Processing High-Speed Data Streams', Springer.

Kleinberg, J. (2002), Bursty and hierarchical structure in streams, *in* 'SIGKDD02'.

Phua, C., Lee, V., Smith-Miles, K. & Gayler, R. (2005), 'A comprehensive survey of data mining-based fraud detection research', *Artificial Intelligence Review*.

Phua, C., Lee, V., & Smith-Miles, K. (2006), 'The personal name problem and a recommended data mining solution', *Encyclopedia of Data Warehousing and Mining (2nd Edition)*.

Phua, C., Lee, V., Gayler, R., & Smith-Miles, K. (2006a), Temporal representation in spike detection of sparse personal identity streams, *in* 'PAKDD06 Workshop on Intelligence and Security Informatics'.

Phua, C., Gayler, R., Smith-Miles, K. & Lee, V. (2006b), Communal detection of implicit personal identity streams, *in* 'IEEE ICDM06 Workshop on Mining Evolving and Streaming Data'.

Phua, C., Gayler, R., Lee, V. & Smith-Miles, K. (2006c), 'On the communal analysis suspicion scoring for identity crime in streaming credit applications', *European Journal of Operational Research*.

Phua, C., Lee, V., Smith-Miles, K. & Gayler, R. (2007), Adaptive communal detection in search of adversarial identity crime, *in* 'ACM SIGKDD07 Workshop on Domain-Driven Data Mining'.

Schneier, B. (2003), *Beyond fear: thinking sensibly about security in an uncertain world*, Copernicus, New York.

Winkler, W. (2006), 'Overview of record linkage and current research directions', Statistical Research Division, U.S. Census Bureau Publication, RR 2006-2.

Wong, W., Moore, A., Cooper, G. & Wagner, M. (2003), Bayesian network anomaly pattern detection for detecting disease outbreaks, *in* 'ICML03', pp. 217–223.

Yoshida, K., Adachi, F., Washio, T., Motoda, H., Homma, T., Nakashima, A., Fujikawa, H., & Yamazaki, K. (2004), Density-based spam detector, *in* 'SIGKDD04', pp. 486–493.