EVALUATING MACHINE LEARNING METHODS AND OVERSAMPLING ON ANOMALY DETECTION USING ELECTRONIC PATIENT RECORD DATA

NICOLE DE VRIES

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY
AT THE SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
OF TILBURG UNIVERSITY

STUDENT NUMBER

2029787

COMMITTEE

dr. Boris Čule
dr. Nevena Rankovic

LOCATION

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science &
Artificial Intelligence
Tilburg, The Netherlands

DATE

January 13, 2023

WORD COUNT

7864

ACKNOWLEDGMENTS

I would like to thank Mark van der Ham, my colleague at PinkRoccade Cloud Solutions, for helping me understand the data and providing insight on the importance of anomaly detection for hospitals.

EVALUATING MACHINE LEARNING METHODS AND OVERSAMPLING ON ANOMALY DETECTION USING ELECTRONIC PATIENT RECORD DATA

NICOLE DE VRIES

Abstract

This thesis aims to investigate the effectiveness of machine learning methods on anomaly detection for failure prediction using the system logs of the integrated ICT-system of a hospital: the Electronic Patient Record (EPR). The findings of this thesis help stakeholders in hospitals to assess which method is efficient in detecting performance problems to avoid failure of an EPR, and help researchers in academic context to assess which method is efficient in correctly classifying anomalies using the system logs of an EPR. The Support Vector Machine (SVM) and Random Forest (RF) are evaluated, as well as the effect of oversampling on these methods and its one-class variations. Synthetic Minority Over-sampling Technique (SMOTE) is used as an oversampling method. Previous studies found that oversampling and the one-class variations outperform the SVM and RF on similar anomaly detection problems. This thesis is the first to evaluate machine learning methods on anomaly detection using the system logs of an EPR for failure prediction. Additionally, this thesis is the first to compare the SVM, RF, oversampling of these models using SMOTE and its one-class variations with each other on an anomaly detection problem. Contradicting previous studies, this thesis finds that, generally, oversampling and the one-class variations did not outperform the SVM and RF. The RF was able to classify all samples in the data perfectly. However, the SVM was unable to identify all anomalies, something the SMOTE-SVM was able to do. The results of this thesis indicate that the RF is the most effective on anomaly detection using the system logs of an EPR.

1 DATA SOURCE/CODE/ETHICS STATEMENT

The dataset used in this thesis was provided by an external source. The author of this thesis complies to the terms and conditions set by the owner of this dataset. The original owner of the data used in this thesis retains ownership of the data during and after the completion of this thesis. The author of this thesis acknowledges that they do not have any legal claim to this data.

The code used in this thesis is not publicly available.

2 INTRODUCTION

2.1 *Societal Relevance*

Thousands of patients were unable to receive healthcare at Maastricht University Medical Center+ on September 8 due to an ICT-related failure (NU.nl, 2022). Luckily emergency care could still be provided, whereas at the Isala hospital in Zwolle earlier last year emergencies had to be diverted to other hospitals due to an ICT-related failure (van Veldhuizen, 2022). ICT failures were the second-most cause of internal hospital crises and disasters in the last two decades in the Netherlands (Klokman et al., 2021). According to the Dutch Safety Board (2020), proper and safe patient care is dependent on ICT. Nearly all hospitals in the Netherlands work with an integrated ICT-system called the Electronic Patient Record (EPR) (Wilman, 2022). As almost all work processes are executed in the EPR, failures can lead to disruption of the workflow in a hospital.

According to Hoover (2016), however, the benefits of the EPR outweigh the drawbacks. These benefits are easy accessibility of information, increased efficiency in workflow and patient care and facilitation of internal cooperation (King, Patel, Jamoom, & Furukawa, 2014; Priestman et al., 2018; van der Graaf, 2012). As such, there is much to gain by early detection of abnormal, or anomalous, behaviour of an EPR. Hospitals have expressed a need for detecting this anomalous behaviour. When anomalies are detected, important stakeholders can take fast measures to avoid further escalation. Summarising, anomaly detection can lead to fast and adequate measures to avoid failures (Dutch Safety Board, 2020).

2.2 *Scientific Relevance*

According to Barrows Jr and Clayton (1996) "anomaly detection depends on unusual behavior or unusual use of system resources (...)". Design and development of anomaly detection methods can be used for predictive

and proactive maintenance and are crucial to reduce the chance of unexpected failure of a system (Fahim & Sillitti, 2019). Anomaly detection algorithms can be used to monitor system usage logs of an EPR to identify malfunctions before patients are affected (Sittig, Lakhani, & Singh, 2022).

Anomaly detection algorithms have been applied on several problems using data from an EPR such as classifying workflows (Boddy, Hurst, Mackay, & El Rhalibi, 2019; Yeng, Fauzi, & Yang, 2020), cyber-attack detection (McGlade & Scott-Hayward, 2019) and medical decision making (Kassakian, Yackel, Gorman, & Dorr, 2017; Ray, McEvoy, Aaron, Hickman, & Wright, 2018). However, there are no studies that use the system logs of EPRs for failure prediction. This leaves an interesting opportunity for research.

Anomaly detection problems can be considered as an extreme class imbalance problem, as anomalies are usually not frequent in the data (Kong, Kowalczyk, Menzel, & Bäck, 2020). Classifiers like the Support Vector Machine (SVM) and Random Forest (RF) show robust performance on several anomaly detection problems (Anton, Kanoor, Fraunholz, & Schotten, 2018; Brown & Mues, 2012; Luo, Pan, Wang, Ye, & Qian, 2019).

Another way to handle class imbalance is through the Synthetic Minority Over-sampling Technique (SMOTE) (Chawla, Bowyer, Hall, & Kegelmeyer, 2002). Research on anomaly detection problems shows that the performance of the SVM and RF is improved when SMOTE is introduced (Tan et al., 2019; Wu et al., 2022).

A heavily imbalanced classification problem could also be considered as a one-class classification problem. One-class algorithms often show better performance than their binary variants on various anomaly detection problems (Tsai & Lin, 2021; Xing & Ji, 2018).

As there are no studies on anomaly detection for failure prediction that use system logs of an EPR, there is an interesting opportunity for research to see how the SVM and RF would perform on this problem. In addition, there is an interesting opportunity to see how the introduction of SMOTE and the one-class variations of these models will affect performance.

2.3 *Research Strategy*

Based on the gaps identified in scientific literature, the following research question is established:

> *To what extent does oversampling or the one-class variations affect the performance of Support Vector Machine and Random Forest on anomaly detection using the system logs of an EPR?*

The following sub-questions are established to answer this question:

RQ1 *How do the Support Vector Machine and Random Forest perform on anomaly detection using the system logs of an EPR?*

RQ2 *How does the introduction of oversampling affect the performance of these models on this anomaly detection problem?*

RQ3 *How do the one-class variations of these models perform on this anomaly detection problem?*

There is a possibility that there are anomalies in the dataset that are not labelled as anomalies, but are actually anomalies. If the models predict regular actions as anomalies, or false positives, they will be analysed manually to determine if they are actually anomalies.

2.4 *Brief Overview of Main Findings*

This study found that oversampling helped the SVM in classifying all anomalies as such. Generally, however, oversampling and the one-class variations did not outperform the SVM and RF. The RF classified all samples in the data correctly.

3 RELATED WORK

3.1 *Anomaly Detection using Electronic Patient Records*

The Netherlands was considered an innovator in healthcare automatisation in the last century, with ICT-systems for hospitals developed at a fast rate. These systems were designed for separate departments, each department requiring a different solution (Zwetsloot-Schonk, 2003). This led to the development of an integrated ICT-system: the Electronic Patient Record (EPR) (Wilman, 2022). An EPR contains all medical and administrative information on a patient and can execute tasks related to patient care (Boll, 2006; Michel-Verkerke, Stegwee, & Spil, 2015).

An EPR continuously gathers data on how well it is performing. These system logs can be analysed by relevant stakeholders to discover problems when an EPR is not functioning properly. This a very reactive manner of dealing with malfunctions, however, which can result in necessary action to be taken too late. According to Sittig et al. (2022) anomaly detection algorithms can be used to monitor system usage logs of EPRs to identify malfunctions before patients are affected.

Anomaly detection has been applied using system logs of EPRs in several fields. Boddy et al. (2019) use Local Outlier Factor Analysis in analysing workflows to detect if no records are accessed without legitimate

access rights. Yeng et al. (2020) use nine different machine learning methods on the same problem, and found that Logistic Regression and Random Forest (RF) performed the best. McGlade and Scott-Hayward (2019) test the K-Nearest Neighbours, Naive Bayes and Support Vector Machine (SVM) on cyber-attack detection and found that the SVM performed the best. Kassakian et al. (2017); Ray et al. (2018) use manual and statistical analysis for anomaly detection in medical decision making.

There are currently no known studies that use anomaly detection methods for failure prediction using the system logs of an EPR. This leaves an interesting opportunity for research to test what anomaly detection models would be appropriate for this problem.

3.2 *Supervised Anomaly Detection*

As anomalies are often not labelled in the data, anomaly detection problems usually require unsupervised machine learning techniques (Hastie, Tibshirani, & Friedman, 2009). However, sometimes anomalies are labelled in the data and supervised methods can be applied (Görnitz, Kloft, Rieck, & Brefeld, 2013). Anomaly detection problems can be considered as an extreme class imbalance problem, as anomalies are usually not frequent in the data (Kong et al., 2020).

The SVM is commonly applied to imbalanced datasets. This algorithm was first introduced by Cortes and Vapnik (1995): features are non-linearly mapped to a high-dimension feature space and through this space a linear decision boundary is drawn. The authors promise high generalisation. Q. Fan, Wang, Li, Gao, and Zha (2017) test multiple variations of the SVM on around thirty different datasets, ranging from little imbalanced to heavily imbalanced. All variations of the SVM showed a fairly robust performance, even on heavily imbalanced datasets. However, the authors found that the SVM does have a tendency to bias towards the negative class as it treats both classes with equal importance. The authors do not compare the variations to other machine learning methods.

The RF is another commonly used algorithm applied on imbalanced datasets. The foundation of this model lies in decision trees: certain rules partition the data for classification (Myles, Feudale, Liu, Woody, & Brown, 2004). RF was first introduced by Ho (1995) as a solution for the difficulty of decision trees generalising to unseen data. In an RF, multiple trees are constructed in randomly selected subspaces of the total feature space. This idea was further developed by Breiman (2001) into the algorithm that is commonly used today. Luo et al. (2019) test a Logistic Regression and RF on multiple imbalanced datasets, and show that RF performs better overall. Brown and Mues (2012) compares the RF to nine other machine learning

algorithms on a credit scoring classification problem using several datasets. The authors find that the RF has a robust performance, even on extremely imbalanced datasets.

Eltanbouly, Bashendy, AlNaimi, Chkirbene, and Erbad (2020); Omar, Ngadi, and Jebur (2013) found in their literature search that the SVM obtains good results in different anomaly detection problems when compared to several machine learning and even deep learning algorithms. However Omar et al. (2013) argue that proper hyperparameter tuning is necessary for obtaining good results.

Pachauri and Sharma (2015) compared RF to the J48 and k-Nearest Neighbour algorithm and found that RF performed the best on anomaly detection in medical wireless sensor networks. Gulenko, Wallschläger, Schmidt, Kao, and Liu (2016) compared RF to 12 other machine learning methods and found that RF performed best in cloud availability anomaly detection. According to Eltanbouly et al. (2020), however, the RF has trouble detecting novel anomalies that are not included in the training set.

There are several studies in literature that compare the RF and SVM on anomaly detection problems using system logs. Al Ali, Svetinovic, Aung, and Lukman (2017) find that both RF and SVM perform similarly and highly effective on malware detection when compared to five other machine learning algorithms, with the RF performing the best. Anton et al. (2018) compare the SVM, RF, k-Nearest Neighbours and Naive Bayes on network traffic anomaly detection. The authors find that both the SVM and RF perform well on this problem, with the SVM outperforming the RF. Timčenko and Gajin (2018) compare the SVM and RF on a similar problem. They, however, find that the RF outperforms the SVM. Abraham et al. (2018) compare the RF, SVM, Naive Bayes, Logistic Regression and Neural Network on anomaly detection for network intrusion. They find that SVM and RF perform the best out of all the algorithms, with the RF showing the best performance. Noh and Basri (2021) compare the SVM and RF in phishing detection, and find that the RF performs the best.

As the SVM and RF perform well on several anomaly detection problems using system logs, it would be interesting to see how these supervised models would perform on anomaly detection for failure prediction using the system logs of an EPR.

One important risk of using supervised methods is that they cannot detect anomalies in the data that have not been labelled as such but are considered anomalies (Aggarwal, 2017; Han, Hu, Huang, Jiang, & Zhao, 2022).

3.3 *Oversampling in Anomaly Detection*

One way to handle imbalanced datasets is through oversampling: new samples are added to the minority class to increase the size of this class (Shelke, Deshmukh, & Shandilya, 2017). The random oversampling technique is one of the earliest proposed techniques (Ling & Li, 1998). This technique duplicates minority samples. An often used oversampling technique is the Synthetic Minority Over-sampling Technique (SMOTE), developed by Chawla et al. (2002). Rather than duplicating samples, SMOTE generates synthetically new instances to provide an algorithm with new information. As random oversampling is prone to models overfitting, SMOTE is a preferred technique among researchers (Sharma, Gosain, & Jain, 2022). Cervantes, Garcia-Lamont, Rodríguez-Mazahua, and Lopez (2020) propose SMOTE as a solution to have the SVM perform more accurately due to this model showing bias to the majority class, treating the minority class as noise. Alraddadi, Lago-Fernández, and Rodríguez (2021) argue that both the RF and SVM benefit in accuracy from introducing SMOTE in anomaly detection problems. According to Fernández, Garcia, Herrera, and Chawla (2018), however, SMOTE has no benefit if samples in classes largely overlap each other.

Gosain and Sardana (2017) test the SVM on six different imbalanced datasets, and show that the performance of the SVM increases on all datasets with SMOTE. Mathew, Pang, Luo, and Leong (2017) test the SVM and SMOTE-SVM on around fifty different datasets and find that generally the SMOTE-SVM performs better. However, on some datasets the SVM outperforms the SMOTE-SVM. There is no comprehensive literature review available which compares the SMOTE-RF and RF on different imbalanced datasets.

Regarding anomaly detection problems using system logs, most literature was found applying SMOTE on network intrusion detection problems using security logging. Alfrhan, Alhusain, and Khan (2020) found that the SVM has a high performance, but achieves perfect performance when SMOTE is introduced. Pajouh, Dastghaibyfard, and Hashemi (2017) use SMOTE before testing SVM and RF. However, they do not compare results of the oversampled dataset to the regular dataset. Tan et al. (2019) show that the performance of RF increases when SMOTE is introduced. Tesfahun and Bhaskari (2013) compare the RF with SMOTE-RF on five different intrusion detection datasets, but found that on only one set SMOTE-RF performance improved. On the other datasets, the performance stayed the same.

When considering earlier research measuring the effect of SMOTE on the SVM and RF, the results differ. However, as SMOTE causes increase

in performance on several anomaly detection problems using system logs, this method is worth considering for the established problem in this study.

3.4 *One-Class Methods in Anomaly Detection*

According to Tsai and Lin (2021), a heavily imbalanced two-class classification problem is similar to a one-class classification problem. Rather than using binary classification algorithms as described in Section 3.2, one-class classification algorithms can be applied to this problem (Xing & Ji, 2018). These methods assume that all samples in the data belong to one class. When new samples are introduced they are given an anomaly score that indicates how far they deviate from the normal data. According to Bellinger, Sharma, and Japkowicz (2012) the performance of one-class classifiers is robust as imbalance increases in the dataset, whereas performance of binary classifiers is not robust if imbalance increases. The One-Class Support Vector Machine (OCSVM) is the unsupervised version of the SVM, specifically designed by Schölkopf, Platt, Shawe-Taylor, Smola, and Williamson (2001) as an extension the SVM in case of unlabelled data. The Isolation Forest (IF) is the unsupervised version of RF and was designed by Liu, Ting, and Zhou (2008). Using a forest-like structure like the RF, the IF isolates anomalies.

One-class classification methods are generally designed for anomaly detection problems (Japkowicz, Myers, Gluck, et al., 1995). These are unsupervised methods, but can be turned into supervised ones when labels are known. One major disadvantage of using these as supervised methods, as mentioned in section 3.2, is that it removes the ability of the models to detect anomalies in the data that have not been labelled as such but are considered anomalies.

As OCSVM and IF are generally used for unlabelled data, there is not much research available that compares these algorithms to its binary variant. Tsai and Lin (2021) found that the IF and OCSVM are especially effective on heavily imbalanced datasets. On less imbalanced datasets, the SMOTE-RF and SMOTE-SVM perform better. However, the authors do not compare the results to a regular SVM and RF.

No comprehensive literature survey could be found that compares the SVM and OCSVM on imbalanced datasets or anomaly detection problems. Liu et al. (2008) compares the IF to the RF, Local Outlier Factor and a distance-based method called ORCA on twelve different anomaly detection datasets. The authors find that IF outperforms the other methods on ten datasets with high performance, with ORCA showing better performance on the other two.

Regarding anomaly detection, Hejazi and Singh (2013) compare the OCSVM to the SMOTE-SVM and SVM on an imbalanced credit card fraud detection, and found that the OCSVM performed the best. S. Fan, Liu, and Chen (2017) compares the IF, OCSVM, RF and SVM on anomaly detection for bankruptcy prediction. They find that the one-class variations outperform the binary classifiers, with IF showing the best performance.

No literature studies have been found that compare SVM and RF to its one-class classification variations on anomaly detection problems using system logs. This study will be the first attempt to do so. As the OCSVM and IF outperform its binary variants on several anomaly detection problems, it would be interesting to consider them.

This study will also be the first to compare the SVM, RF, SMOTE-SVM, SMOTE-RF, OCSVM and IF together.

4 METHODS

4.1 *Support Vector Machine*

The distance between the soft margins is affected by regularisation hyperparameter C. A larger value of C will lead to a smaller margin, whereas a smaller value will lead to a larger margin. One concern of having a small margin is overfitting: the SVM will capture all the variance, or unique features, in the training data but will not generalise to the test data (Hsu, Chang, Lin, et al., 2003).

A kernel function maps all samples in a higher dimension. If the data is linearly separable, a linear kernel function can be used. In case of a nonlinear separable dataset, the dataset needs to be transformed using different kernel functions (Suthaharan, 2016). A detailed description of these kernel functions can be found in Appendix A (page 31). All nonlinear kernel functions use the value of hyperparameter gamma: the larger its value, the more linear the decision boundary will be.

4.2 *Random Forest*

Breiman (2001) provides the following definition: "A random forest is a classifier consisting of a collection of tree-structured classifiers {h(x, θ_k), k = 1, . . .} where the {θ_k } are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input x". The most popular class is decided through a simple majority vote.

According to Probst, Wright, and Boulesteix (2019) the default parameters of the RF usually provide good results. However, the authors argue that if misclassification comes at high cost it is worth it to tune the hy-

perparameters to see if performance increases. The first hyperparameter is the number of trees in the forest: more trees usually results in better performance (Probst & Boulesteix, 2017). The second is the maximum depth of each tree. The third is the maximum amount of features used by each tree: including more features may lead to better performance (Shreyas, Akshata, Mahanand, Shagun, & Abhishek, 2016).

4.3 *SMOTE*

A new, synthetic, minority sample is created using Equation 1:

$$p_i = x_i + rand(0,1) * (m_i - x_i) \tag{1}$$

For each minority sample, x_i, the difference to its k nearest neighbours of minority samples is calculated, with m_i being a random nearest neighbour of k (Chawla et al., 2002; Wang, Dai, Shen, & Xuan, 2021). The difference between x_i and m_i is then multiplied by a random number, uniformly distributed between 0 and 1. This is then added to the minority sample. This will lead to a new, synthetic, sample along the line segment of two minority samples.

4.4 *One-Class Support Vector Machine*

The goal of the OCSVM is to map all samples to feature space ϕ (Schölkopf et al., 2001). This is done through using a kernel function, k, as shown in Equation 2

$$k(x, y) = (\phi(x) \cdot \phi(y)) \tag{2}$$

A function, f, gives all training samples the value of +1 in the feature space. Training samples are mapped corresponding to k, and then separated from the origin space using a hyperplane with a maximum margin. For each new sample x, $f(x)$ will determine which side of the hyperplane this sample will fall on. +1 indicates a regular sample, -1 indicates an anomaly.

To separate the data from its origin space, the following quadratic program is used:

$$\min_{w,\xi,\rho} \frac{1}{2}\|w\|^2 + \frac{1}{v\ell}\sum_{i=1}^{n}\xi_i - \rho \tag{3}$$

$$\text{subject to} : (w \cdot \phi(x_i)) \geq \rho - \xi_i, \xi_i \geq 0 \tag{4}$$

Where w is a normal weight vector, ρ is a bias parameter and ξ_i is a nonzero slack variable that allows some anomalies to lie on the wrong side on the hyperplane. $||w||$ is used to maximise the margin. The cost of misclassification is controlled by v: this regularisation parameter is the upper bound on the fraction of margin errors and the lower bound of the fraction of support vectors. ξ_i is penalised in the objective function, however, so the decision function becomes as follows:

$$f(x) = sgn((w \cdot \phi(x)) - \rho) \tag{5}$$

Using multipliers α_i and $\beta_i \geq 0$, a Lagrangian function is formulated as seen in Equation 6

$$L(w, \xi, \phi, \alpha, \beta) = \frac{1}{2}\|w\|^2 + \frac{1}{v\ell}\sum_{i=1}^{n}\xi_i - \rho - \sum_{i=1}^{n}\alpha_i((w \cdot \phi(x_i)) - \rho + \xi_i) - \sum_{i=1}^{n}\beta_i\xi_i \tag{6}$$

The derivatives of variables w, ξ, ρ are minimised using Equation 6. This allows for the creation of a dual problem in which all variables have low dimensions. The x_i that has a corresponding $\alpha_i > 0$ will be used as a support vector. The addition of support vectors changes decision function 5 into a kernel expansion:

$$f(x) = sgn(\sum_{i=1}^{n}\alpha_i k(x_i, x) - \rho \tag{7}$$

Where ρ is defined as follows:

$$\rho = w \cdot \phi(x_i)) = \sum_j \alpha_j k(x_j, x_i) \tag{8}$$

And multipliers α as follows:

$$\sum_{i=1}^{n}\alpha_i = 0 \leq \alpha_i \leq \frac{1}{vn} \tag{9}$$

Summarising Equation 7 maps all samples to the feature space. Positive samples are considered regular samples, negative samples are considered anomalies.

4.5 *Isolation Forest*

Similar to a regular decision tree, samples are partitioned recursively until all samples are isolated. Liu et al. (2008) found that the partitioning path of trees is noticeably shorter for anomalies as there are fewer instances of these in the data. Additionally, anomalies have distinguishable feature values and are therefore more likely to be separated early. Therefore, when a forest of these trees collectively produce short path lengths for some particular samples, it is very likely that these instances are anomalies.

Liu et al. (2008) give the following definition for an Isolation Tree (IT): "Let T be a node of an isolation tree. T is either an external-node with no child, or an internal-node with one test and exactly two daughter nodes (T_l,T_r). A test consists of an attribute q and a split value p such that the test $q < p$ divides data points into T_l and T_r".

An anomaly score is calculated from each IT using path length *h(x)*, where x is a sample from the dataset. The average path length of an IT is calculated using Equation 10.

$$c(n) = 2H(n-1) - (2(n-1)/n) \tag{10}$$

Where n is the number of external nodes, *c(n)* is the average of *h(x)* given n. The anomaly score, s, of x is calculated using Equation 11. This equation outputs a value between 0 and 1.

$$s(x,n) = 2 - \frac{E(h(x))}{c(n)} \tag{11}$$

Where *E(h(x))* is the average of *h(x)* from a collection of ITs. If *E(h(x))* is (close to) 0 this indicates a short average path length, meaning the anomaly score will be (close to) 1 and vice versa. If the anomaly score is 0.5, the entire sample does not have a distinct anomaly.

There are two hyperparameters that can be tuned for the IF: the number of trees, similar as for the RF described in Section 4.2, and the sub-sampling size. According to Liu et al. (2008) sub-samples make the cluster of normal training samples smaller, thus making it easier to identify anomalies.

5 EXPERIMENTAL SETUP

Figure 1 illustrates the research methodology using a flowchart. The following subsections provide in-depth information on the data preprocessing, used methods and evaluation metrics. Appendix B (page 32) provides a summary of the coding environment, including all used libraries.

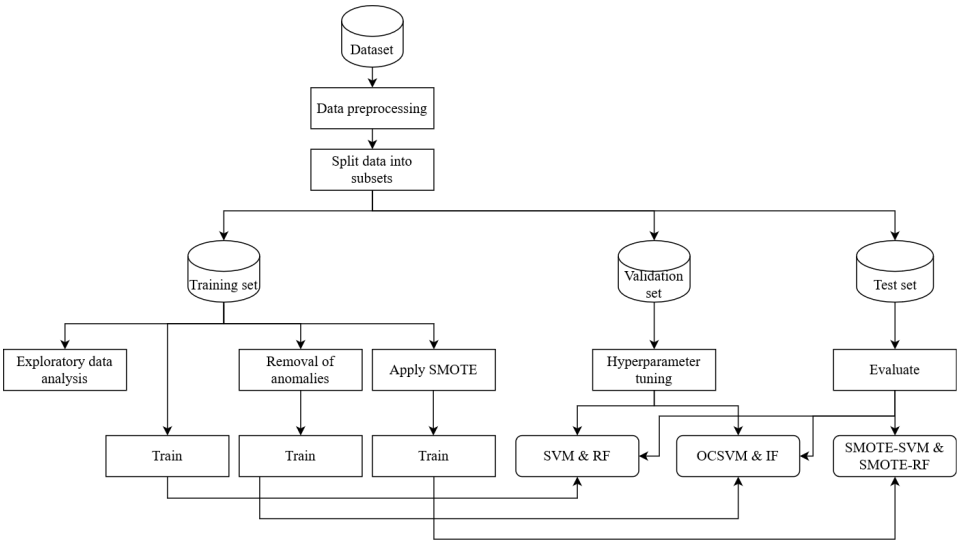

Figure 1: Flowchart illustrating the research methodology used in this thesis.

5.1 *Dataset Description*

Systems logs of one day in September were retrieved from an EPR from one of the largest hospitals in the Netherlands. These system logs contain information about actions users have performed in this EPR. The data consists of 601,861 actions: 587,943 are regular workflow actions and 13,918, or 2.3%, are anomalies. The dataset is thus heavily imbalanced. Anomalies are labelled as such based on four features, as seen in Table 1. If a feature value of an action is larger than the threshold, this action is considered anomalous. An anomaly can have larger values than the threshold on multiple features.

Table 1: The four features used in the dataset. Table contains the name of the features, a description of the features and the threshold set to determine if a sample is anomalous.

Feature Name	Feature Description	Threshold
Total time	The time in seconds it took for an entire action to execute	15
Query time	The time in seconds it took for the queries within an action to execute	6
Query count	Total number of queries executed in an action	2,000
Query row count	Total number of rows the queries retrieved from the database	20,000

5.2 *Data Preprocessing*

No values were missing in the dataset. The dataset contains two class labels: action, indicating a regular action and detail, indicating an anomalous

action that is causing performance problems. These labels are binarised as most machine learning methods require numerical values for classification (Dahouda & Joe, 2021). Regular actions are given the value of 0, anomalies the value of 1.

The dataset was split into three different subsets for the implementation of the methods: a training set consisting of 70% of the data, a validation set consisting of 15%, and a test set consisting of the remaining 15%. Because the dataset is imbalanced, stratified sampling is implemented to ensure that the relative class frequencies are preserved in each set (Bennett & Carvalho, 2010). All samples are shuffled in each set. The random state was set to 196, ensuring reproducible data splits. The data is standardised using the corresponding means and variances of all features to ensure faster training times of the SVM and OCSVM (Ben-Hur, Ong, Sonnenburg, Schölkopf, & Rätsch, 2008).

SMOTE was only applied to the training set to ensure the validation and test stay independent. The random state was set to 196, to ensure each sample will be the same if reproduced. The application of SMOTE results in a balanced training set of 823,118 samples, consisting 50% out of regular actions and 50% out of anomalies.

The one-class variations, OCSVM and IF, only use regular actions for training. Anomalies are thus removed from the training set for these models, but are included in the validation and test set for hyperparameter tuning and evaluation. Regular actions are given the value of 1 and anomalies the value of -1 to allow for evaluation,

5.3 *Exploratory Data Analysis*

Table 2: Descriptive statistics for regular actions, rounded at two decimals. Time in seconds.

	Condition			
	Total Time	Query Time	Query Count	Query Row Count
Mean	0.98	0.27	103.00	736.67
Std	1.46	0.64	226.74	1932.58
Min	0.00	0.00	0.00	0.00
Median	0.35	0.03	10.00	23.00
Max	15.00	6.00	2000.00	19995.00

Exploratory data analysis was performed only on the training set, consisting of 421,439 actions, to ensure the validation and test set stay independent and no information is inferred from them. Out of these

Table 3: Descriptive statistics for anomalies, rounded at two decimals. Time in seconds.

	Condition			
	Total Time	Query Time	Query Count	Query Row Count
Mean	11.92	8.13	1770.02	18822.56
Std	19.60	15.51	1952.75	39985.83
Min	0.56	0.00	0.00	0.00
Median	8.96	5.57	1671.00	6564.00
Max	1081.97	612.66	85878.00	2434774.00

actions 9,743 are anomalies, or 2.3%. This means that stratification of the labels worked as intended.

Descriptive statistics for the regular actions in the training set can be inferred from Table 2. The maximum values of the features do not pass the thresholds as described in Section 5.1. The minimum value of total time is 0, which is rare but possible: an action could complete instantaneously. Regarding the features concerning queries, it is possible that an action is only using CPU resources and is not retrieving anything from a database. From the median and mean can be inferred that all features generally have low values. The standard deviation suggests that there is not much variability around the time features, more so for the count features.

Table 3 contains descriptive statistics for anomalous actions in the training set. The minimum values are zero, with the exception of total time which has the minimum value of around half a second. The mean and median indicate that overall values are larger when compared to the descriptive statistics of regular actions. The standard deviation indicates that there is a reasonable amount of variability in the data for all features.

Figure 2 plots all features in the dataset against one another. From this plot can be inferred that the features duration and total query duration have a strong correlation for anomalous actions: if duration increases, total query duration increases and contrariwise. Other features do not seem to have a clear correlation between them. From this figure can also be inferred that regular actions concentrate around smaller feature values, while anomalous actions take larger values.

The application of SMOTE on the training set resulted in 823,118 actions in the dataset, a 50% split of actions and anomalies. Based on the descriptive statistics (found in Appendix C, page 33) can be inferred that the distribution of the anomalies differs very little from the dataset used for the SVM, RF and its one-class variations.

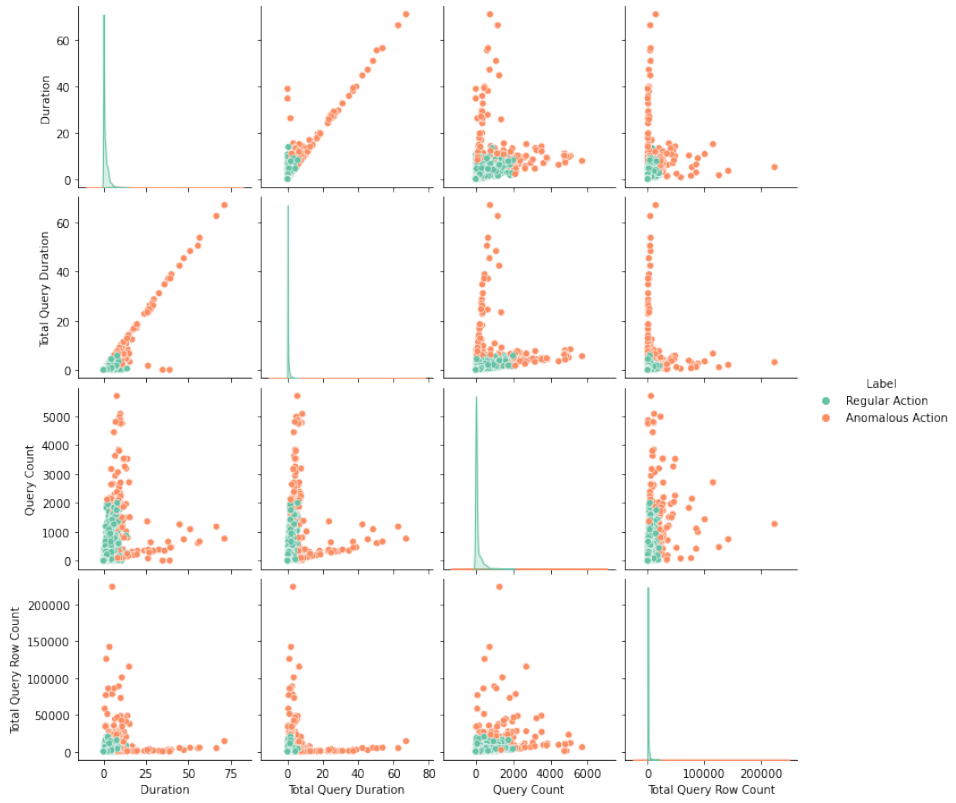

Figure 2: Pairplot visualising the relationship between the four features in the data. A subset of 10,000 random samples were taken from the training set to guarantee legibility of the plot. The diagonal plots the Kernel Density Estimates of the features, corresponding to the density function of each feature.

5.4 *Hyperparameter settings*

For each method, a grid search function is defined which tests all hyperparameter combinations on the validation data (Yu & Zhu, 2020). The hyperparameter combination with the highest F_1 score will be used on the test data. The F_1 score of each combination for each model can be found in Appendix D (page 34).

5.4.1 *Support Vector Machine*

All four kernel methods of the SVM will be tested. These are the linear, Radial Basis Function, polynomial and sigmoid function. The value of C and gamma need to be tuned as well. As there is no similar research available, common used values of these hyperparameters will be tested on the validation set (Hsu et al., 2003). These are the gamma values of 0.1, 0.01, 0.001 and C values of 0.1, 1, 10 and 100.

The final model uses a C value of 100, gamma value of 0.1 and Radial Basis Function as the kernel function.

5.4.2 *Random Forest*

Two hyperparameters are tuned, using the research by Shreyas et al. (2016) as a baseline. The number of trees is increased in steps of 100, starting at 100 and ending at 500. The default value of maximum features used is set to two, this number is usually increased in steps of two. As there are four features in the dataset, the values of two and four were tested. Maximum depth will not be tuned for this model, as there are only four features in the dataset: the tree will expand until its nodes are pure.

The final model uses 200 trees and all four features.

5.4.3 *One-Class Support Vector Machine*

The same hyperparameters will be tested for the OCSVM as for the SVM, explained in Section 5.4.1. Instead of C, hyperparameter v as explained in Section 4.4 will be tuned. There is no common strategy for tuning the value of v, but its value has to be between 0 and 1. The work by Eude and Chang (2018) is followed and the value is increased in a step size of 0.02, starting at 0.01 and ending at 0.09.

The final model uses a v value of 0.01, gamma value of 0.001 and Radial Basis Function as the kernel function.

5.4.4 *Isolation Forest*

The number of trees will be increased similarly as done for the RF, described in Section 5.4.2. The creators of the model argue that 256 is the minimum number of samples needed to accurately inform the model. This number is increased through the power of two until the maximum number of samples in the training set is reached. According to the authors, sub-samples make the cluster of normal training samples smaller, thus making it easier to identify anomalies. However, the defined grid-search found that using all samples caused the best performance.

The final model therefore uses all samples in the dataset, paired with 100 trees.

5.5 *Evaluation Method*

Recall will be used as a performance metric (Powers, 2011). Recall is the fraction of positives that are correctly classified: it measures what proportion of true anomalies was identified as such. Precision will be used

as a metric as well. The precision is the fraction of true positives divided by the total number of positive prediction: it measures what proportion of identified anomalies are true anomalies.

The F_1 score, the harmonic mean between precision and recall, will be used to assess the overall performance of each model (Powers, 2011). Confusion matrices are also generated for each model, these will give insight on how the models confuse both classes (Susmaga, 2004).

6 RESULTS

6.1 *Performance of Each Model*

In this section, classification performance of all models on the test dataset will be described. Table 4 contains the evaluation scores of all models. The recall scores of every model were (near-)perfect. The SMOTE variations of the SVM and RF did not outperform the SVM and RF. The IF achieved a perfect recall score like the RF. The precision scores of the OCSVM and IF are lower than the other models, resulting in a lower F_1 score for these models. The IF was the worst performing model overall based on precision and F_1 score. Based on recall, the OCSVM is the worst performing model. The RF was the best performing model, achieving a perfect score on all evaluation metrics.

The confusion matrices of the models (see Figure 3 up to and including Figure 8) show that the SMOTE-SVM, OCSVM and IF sometimes predicted regular actions as anomalies, or false positives, explaining the lower precision scores and F_1 scores of these models.

Table 4: Performance of all models on the test data using the evaluation metrics of precision, recall and F_1 score. Rounded at three decimals. SMOTE = Synthetic Minority Over-sampling Technique. Highest scores in bold.

Models	Precision	Recall	F1 Score
Support Vector Machine	0.995	0.989	0.992
Random Forest	**1.000**	**1.000**	**1.000**
SMOTE-Support Vector Machine	0.889	**1.000**	0.941
SMOTE-Random Forest	0.999	0.999	0.999
One-Class Support Vector Machine	0.779	0.989	0.872
Isolation Forest	0.362	**1.000**	0.531

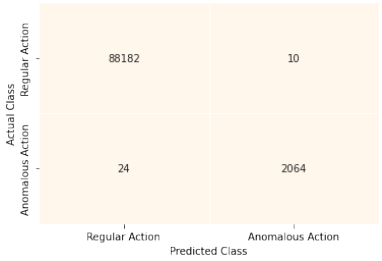

Figure 3: Confusion matrix of the Support Vector Machine.

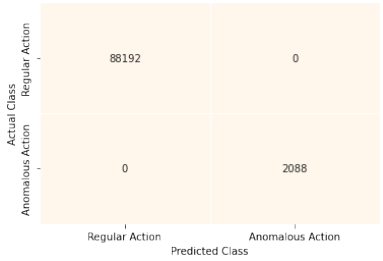

Figure 4: Confusion matrix of the Random Forest.

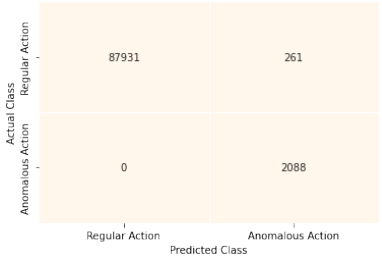

Figure 5: Confusion matrix of the Synthetic Minority Over-Sampling Method-Support Vector Machine.

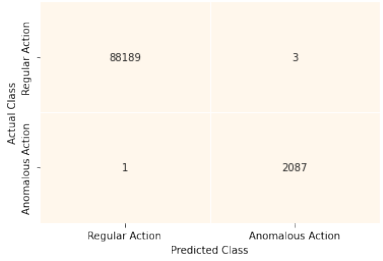

Figure 6: Confusion matrix of the Synthetic Minority Over-Sampling Method-Random Forest.

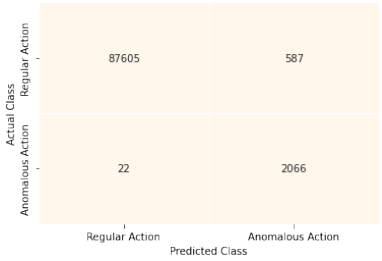

Figure 7: Confusion matrix of the One Class-Support Vector Machine.

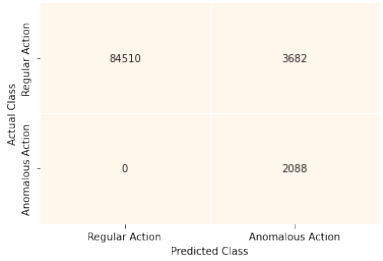

Figure 8: Confusion matrix of the Isolation Forest.

6.2 *False Positives Detected by Each Model*

Each model, with exception of the RF, classified false positives in the dataset (see Appendix E, page 36). Based on manual analysis and the thresholds set as described in Section 5.1, it was concluded that none of the false positives were actually anomalies. For the SVM, SMOTE-SVM and SMOTE-RF the false positives were very close to at least one of the thresholds. The OCSVM and IF classified some false positives that had a feature value close to one of the thresholds, but also classified some false positives that were not as close.

7 DISCUSSION

This study aims to determine whether oversampling or the one-class variations affect the performance of the SVM and RF on anomaly detection using the systems logs of an EPR. It measures performance using the evaluation metrics precision, recall and F_1 score.

The RF was the best performing model, achieving perfect scores on all metrics. Oversampling, done through SMOTE, achieved a near-perfect performance for the RF. The one-class variation of the RF, the IF, achieved perfect recall, meaning it was able to identify all anomalies in the data as such. However, the IF classified a large amount of regular actions as anomalies, resulting in a low precision value and thus a lower overall score.

The SVM achieved a high performance overall, but was unable to identify all anomalies as such. Oversampling did classify all anomalies correctly, but classified more regular actions as anomalies as well. This resulted in a lower precision value and a lower performance overall. The one-class variation of the SVM, the OCSVM, failed to outperform the SVM.

Overall, oversampling and the one-class variations did not outperform the SVM and RF based on the F_1 score. However, SMOTE-SVM was able to detect all anomalies in the data, something the SVM was unable to do. The IF was also able to identify all anomalies as such.

7.1 *Interpretation of Results*

The first sub-question was established to see how well the SVM and RF would perform on this anomaly detection problem. The SVM performed well on average, obtaining a near-perfect score for all evaluation metrics. The SVM has a reputation of performing well on imbalanced data and different anomaly detection problems (Eltanbouly et al., 2020; Q. Fan et al., 2017; Omar et al., 2013). However, Omar et al. (2013) argue that proper hyperparameter tuning is necessary for obtaining good results. This was established through an extensive grid-search. The RF also has a reputation of performing well on imbalanced data and several anomaly detection problems (Brown & Mues, 2012; Gulenko et al., 2016; Luo et al., 2019; Pachauri & Sharma, 2015). When the SVM and RF are compared on anomaly detection problems using system logs the general results are that the RF outperforms the SVM, which is in line with the findings of this study (Abraham et al., 2018; Al Ali et al., 2017; Noh & Basri, 2021; Timčenko & Gajin, 2018). An explanation could be that the SVM has a tendency to bias towards the negative class (Q. Fan et al., 2017).

The second sub-question was established to detect how oversampling, specifically SMOTE, would affect the performance of the SVM and RF. This

study found that the SMOTE-SVM had a perfect recall, unlike the SVM, but generally performed worse than the SVM as it classified more regular actions as anomalies. According to Cervantes et al. (2020), SMOTE is a solution when the SVM has a tendency to bias towards the majority class. This may explain why the SMOTE-SVM had a better recall. Mathew et al. (2017) found that the regular SVM performs better on some imbalanced datasets, which is in line with the findings of this study. Regarding anomaly detection using system logs, Alfrhan et al. (2020) found that SMOTE-SVM outperformed the SVM, which contradicts the findings of this study. The SMOTE-RF did not classify each instance perfectly like the RF, but did have near-perfect scores on all evaluation metrics. Studies using system logs for anomaly detection found that the SMOTE-RF has a better performance (Tan et al., 2019), while others find that a regular RF performs better (Tesfahun & Bhaskari, 2013). Alraddadi et al. (2021) argued that both the SVM and RF would benefit in performance when introducing SMOTE, but this study found the opposite to be true when measured in F_1 scores. When classes overlap each other, SMOTE will not increase performance (Fernández et al., 2018). An explanation could be that there are many regular actions in the test set that have feature values near the threshold of being anomalous as defined in Section 5.1, but are not actual anomalies. Through manual analysis could be inferred that the SMOTE-SVM and SMOTE-RF classified regular actions as anomalies with values near the threshold.

The third sub-question was established to detect how the one-class variations of the SVM and RF, OCSVM and IF, would perform on this problem. The OCSVM failed to outperform the SVM. The IF had a perfect recall score, like the regular RF, but had a low precision score resulting in a low F_1 score. This contradicts the work by Bellinger et al. (2012), whom argue that the performance of one-class classifiers are stable with high imbalance, whereas the performance of binary classifiers decreases. There were no studies available which test the OCSVM and IF on anomaly detection problems using system logs, but studies on imbalanced datasets and other anomaly detection problems found that the OCSVM and IF outperform its binary variants (S. Fan et al., 2017; Hejazi & Singh, 2013; Liu et al., 2008). The findings of this study contradict these previous studies.

According to Aggarwal (2017) & Han et al. (2022), one important risk of using supervised methods is that they cannot detect anomalies in the data that have not been labelled as such but are considered anomalies. Therefore, regular actions that were classified as anomalies, or false positives, were analysed manually guided by the thresholds defined in Section 5.1. Possible anomalies that were missed could then be identified. However, it was found that none of the false positives were actual anomalies. All identified false positives had at least one feature value that was close to its threshold for

the SVM, SMOTE-SVM and SMOTE-RF. The OCSVM and IF identified some false positives that had one feature value close to its threshold, but also some that were not close to a threshold. Interestingly, the RF was the only model that did not classify any false positives. This could be because it has trouble detecting novel anomalies that are not included in the training set, as argued by Eltanbouly et al. (2020).

Summarising, the RF, SMOTE-SVM and IF were the only models able to identify each anomaly in the data. The RF has proven to be the best performing model, classifying all samples in the data correctly. Therefore, the RF is an appropriate choice for anomaly detection to prevent failure prediction of an EPR. Even though the other models did not detect any actual anomalies, the RF may not be a good choice if the goal is to identify anomalies that were not labelled as such in the data.

7.2 *Limitations and Recommendations for Future Research*

The OCSVM and IF calculate anomaly scores for each sample identified as anomalies. As such, in this study, all samples with an anomaly score were classified as anomalies. It was beyond the scope of this study to consider thresholds for these anomaly scores. When using these models in an unsupervised manner as these were originally designed to do, one could disregard samples with a low anomaly score and only focus on anomalies with a high score. Herein lies an opportunity for future research, which will likely lessen the amount of false positives generated by the one-class variations.

As mentioned by Aggarwal (2017); Han et al. (2022), a disadvantage of using supervised methods for anomaly detection is that they cannot detect anomalies in the data that have not been labelled as such but are actually considered anomalies. This thesis used two unsupervised methods specifically designed for detecting anomalies Japkowicz et al. (1995) as supervised methods. Through a manual analysis of false positives, potential anomalies that were flagged by each model were reviewed. However, it turned out that these are not actually considered anomalies as they did not cross the thresholds discussed in Section 5.1. It is beyond the scope of this study to add more features to the dataset which could possibly indicate if an action is anomalous and causes performance problems. Herein lies an interesting opportunity for future research. If labels are known, the SVM and SMOTE-SVM would be interesting to consider based on the results of this study. The SVM cannot be applied if the labels are unknown, but it provides an opportunity to test unsupervised classifiers to detect these anomalies. Based on the results of this study, the IF would be a favourable model to test.

The reliability of these results is impacted by the clear set of thresholds used to label an anomaly. Next to making this task less challenging, a different set of thresholds will lead to a different set of anomalies. For example, it is possible that the thresholds defined now are not accurate in the future. This means that the results of this study are biased. Additionally, it is possible that there are anomalies in the dataset that do not fit these thresholds but might actually be anomalies. There lies a possibility in future research to let go of the defined thresholds and use unsupervised methods to detect anomalies. These methods can continuously define thresholds based on current data, eliminating bias.

Finally, this study is the first to test the proposed models on anomaly detection of performance using the system logs of an EPR. Thus, future research is needed to establish whether the results of this study are generalisable. As there were no previous studies available which used the system logs of an EPR, most studies in the related work section (Section 3) focused on different areas of anomaly detection using a similar data format. As this study is also the first to compare the SVM, RF, SMOTE-SVM, SMOTE-RF, OCSVM and IF together, there lies an interesting possibility in testing if the results are reproducible in other areas of research. An example could be network intrusion detection, as Section 3 indicates that this is a common research area for anomaly detection.

8 CONCLUSION

This thesis aimed to determine whether oversampling or the one-class variations affect the performance of the SVM and RF on anomaly detection using the systems logs of an EPR. This study is the first to test machine learning methods on detecting anomalies for failure prediction using the system logs of an EPR. This thesis contributes to existing research by being the first to compare the SVM, RF, SMOTE-SVM, SMOTE-RF, OCSVM and IF on an anomaly detection problem. Performance was measured using the evaluation metrics precision, recall and F_1 score. The results of this study find that, contrary to previous studies on anomaly detection, the SVM and RF outperform the other models on this specific problem. The RF was able to correctly classify all samples in the data. The SMOTE-SVM was able to identify all anomalies in the data, however, something the SVM was unable to. Overall, oversampling and the one-class variations could not outperform the SVM and RF.

The findings of this thesis help stakeholders in hospitals to assess which method is efficient in detecting performance problems to avoid failure of an EPR. Using this information stakeholders can take fast and adequate measures, ensuring that patients have access to proper, safe and necessary

care. Failure detection avoids the disruption of the workflow in hospitals, benefiting the staff of the hospital as well. The findings help researchers in academic context to assess which method is efficient in correctly classifying anomalies using the system logs of an EPR.

REFERENCES

Abraham, B., Mandya, A., Bapat, R., Alali, F., Brown, D. E., & Veeraraghavan, M. (2018). A comparison of machine learning approaches to detect botnet traffic. In *2018 international joint conference on neural networks (ijcnn)* (pp. 1–8).

Aggarwal, C. C. (2017). An introduction to outlier analysis. In *Outlier analysis* (pp. 1–34). Springer.

Al Ali, M., Svetinovic, D., Aung, Z., & Lukman, S. (2017). Malware detection in android mobile platform using machine learning algorithms. In *2017 international conference on infocom technologies and unmanned systems (trends and future directions)(ictus)* (pp. 763–768).

Alfrhan, A. A., Alhusain, R. H., & Khan, R. U. (2020). Smote: Class imbalance problem in intrusion detection system. In *2020 international conference on computing and information technology (iccit-1441)* (pp. 1–5).

Alraddadi, F. S., Lago-Fernández, L. F., & Rodríguez, F. B. (2021). Impact of minority class variability on anomaly detection by means of random forests and support vector machines. In *International work-conference on artificial neural networks* (pp. 416–428).

Anton, S. D., Kanoor, S., Fraunholz, D., & Schotten, H. D. (2018). Evaluation of machine learning-based anomaly detection algorithms on an industrial modbus/tcp data set. In *Proceedings of the 13th international conference on availability, reliability and security* (pp. 1–9).

Barrows Jr, R. C., & Clayton, P. D. (1996). Privacy, confidentiality, and electronic medical records. *Journal of the American medical informatics association*, *3*(2), 139–148.

Bellinger, C., Sharma, S., & Japkowicz, N. (2012). One-class versus binary classification: Which and when? In *2012 11th international conference on machine learning and applications* (Vol. 2, pp. 102–106).

Ben-Hur, A., Ong, C. S., Sonnenburg, S., Schölkopf, B., & Rätsch, G. (2008). Support vector machines and kernels for computational biology. *PLoS computational biology*, *4*(10), e1000173.

Bennett, P. N., & Carvalho, V. R. (2010). Online stratified sampling: evaluating classifiers at web-scale. In *Proceedings of the 19th acm international conference on information and knowledge management* (pp. 1581–1584).

Bisong, E. (2019). Matplotlib and seaborn. In *Building machine learning and deep learning models on google cloud platform* (pp. 151–165). Springer.

Boddy, A. J., Hurst, W., Mackay, M., & El Rhalibi, A. (2019). Density-based outlier detection for safeguarding electronic patient record systems. *IEEE Access*, *7*, 40285–40294.

Boll, M. (2006). Kritieke succesfactoren bij de implementatie van een

elektronisch patiëntendossier. *Universiteit van Tilburg*, 1–142.

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5–32.

Brown, I., & Mues, C. (2012). An experimental comparison of classification algorithms for imbalanced credit scoring data sets. *Expert Systems with Applications*, *39*(3), 3446–3453.

Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., & Lopez, A. (2020). A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, *408*, 189–215.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321–357.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*(3), 273–297.

Dahouda, M. K., & Joe, I. (2021). A deep-learned embedding technique for categorical features encoding. *IEEE Access*, *9*, 114381-114391. doi: 10.1109/ACCESS.2021.3104357

Dutch Safety Board. (2020, feb). *Patiëntveiligheid bij ict-uitval in ziekenhuizen* (Tech. Rep.). Lange Voorhout 9, 2514 EA The Hague: Dutch Safety Board.

Eltanbouly, S., Bashendy, M., AlNaimi, N., Chkirbene, Z., & Erbad, A. (2020). Machine learning techniques for network anomaly detection: A survey. In *2020 ieee international conference on informatics, iot, and enabling technologies (iciot)* (pp. 156–162).

Eude, T., & Chang, C. (2018). One-class svm for biometric authentication by keystroke dynamics for remote evaluation. *Computational Intelligence*, *34*(1), 145–160.

Fahim, M., & Sillitti, A. (2019). Anomaly detection, analysis and prediction techniques in iot environment: A systematic literature review. *IEEE Access*, *7*, 81664–81681.

Fan, Q., Wang, Z., Li, D., Gao, D., & Zha, H. (2017). Entropy-based fuzzy support vector machine for imbalanced datasets. *Knowledge-Based Systems*, *115*, 87–99.

Fan, S., Liu, G., & Chen, Z. (2017). Anomaly detection methods for bankruptcy prediction. In *2017 4th international conference on systems and informatics (icsai)* (pp. 1456–1460).

Fernández, A., Garcia, S., Herrera, F., & Chawla, N. V. (2018). Smote for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. *Journal of artificial intelligence research*, *61*, 863–905.

Görnitz, N., Kloft, M., Rieck, K., & Brefeld, U. (2013). Toward supervised anomaly detection. *Journal of Artificial Intelligence Research*, *46*, 235–

262.

Gosain, A., & Sardana, S. (2017). Handling class imbalance problem using oversampling techniques: A review. In *2017 international conference on advances in computing, communications and informatics (icacci)* (pp. 79–85).

Gulenko, A., Wallschläger, M., Schmidt, F., Kao, O., & Liu, F. (2016). Evaluating machine learning algorithms for anomaly detection in clouds. In *2016 ieee international conference on big data (big data)* (pp. 2716–2721).

Han, S., Hu, X., Huang, H., Jiang, M., & Zhao, Y. (2022). Adbench: Anomaly detection benchmark. *arXiv preprint arXiv:2206.09426*.

Hao, J., & Ho, T. K. (2019). Machine learning made easy: a review of scikit-learn package in python programming language. *Journal of Educational and Behavioral Statistics*, *44*(3), 348–361.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). Unsupervised learning. In *The elements of statistical learning* (pp. 485–585). Springer.

Hejazi, M., & Singh, Y. P. (2013). One-class support vector machines approach to anomaly detection. *Applied Artificial Intelligence*, *27*(5), 351–366.

Ho, T. K. (1995). Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition* (Vol. 1, pp. 278–282).

Hoover, R. (2016). Benefits of using an electronic health record. *Nursing*, *46*(7), 21–22.

Hsu, C.-W., Chang, C.-C., Lin, C.-J., et al. (2003). *A practical guide to support vector classification.* Taipei, Taiwan.

Japkowicz, N., Myers, C., Gluck, M., et al. (1995). A novelty detection approach to classification. In *Ijcai* (Vol. 1, pp. 518–523).

Kassakian, S. Z., Yackel, T. R., Gorman, P. N., & Dorr, D. A. (2017). Clinical decisions support malfunctions in a commercial electronic health record. *Applied clinical informatics*, *8*(03), 910–923.

King, J., Patel, V., Jamoom, E. W., & Furukawa, M. F. (2014). Clinical benefits of electronic health record use: national findings. *Health services research*, *49*(1pt2), 392–404.

Klokman, V. W., Barten, D. G., Peters, N. A., Versteegen, M. G., Wijnands, J. J., van Osch, F. H., … Boin, A. (2021). A scoping review of internal hospital crises and disasters in the netherlands, 2000–2020. *PloS one*, *16*(4), e0250551.

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B. E., Bussonnier, M., Frederic, J., … Willing, C. (2016). *Jupyter notebooks-a publishing format for reproducible computational workflows.*

Kong, J., Kowalczyk, W., Menzel, S., & Bäck, T. (2020). Improving imbalanced classification by anomaly detection. In *International conference*

on parallel problem solving from nature (pp. 512–523).

Lemaître, G., Nogueira, F., & Aridas, C. K. (2017). Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *The Journal of Machine Learning Research*, *18*(1), 559–563.

Ling, C. X., & Li, C. (1998). Data mining for direct marketing: Problems and solutions. In *Kdd* (Vol. 98, pp. 73–79).

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation forest. In *2008 eighth ieee international conference on data mining* (pp. 413–422).

Luo, H., Pan, X., Wang, Q., Ye, S., & Qian, Y. (2019). Logistic regression and random forest for effective imbalanced classification. In *2019 ieee 43rd annual computer software and applications conference (compsac)* (Vol. 1, pp. 916–917).

Mathew, J., Pang, C. K., Luo, M., & Leong, W. H. (2017). Classification of imbalanced data by oversampling in kernel space of support vector machines. *IEEE transactions on neural networks and learning systems*, *29*(9), 4065–4076.

McGlade, D., & Scott-Hayward, S. (2019). Ml-based cyber incident detection for electronic medical record (emr) systems. *Smart Health*, *12*, 3–23.

McKinney, W. (2011). pandas: a foundational python library for data analysis and statistics. *Python for high performance and scientific computing*, *14*(9), 1–9.

Michel-Verkerke, M. B., Stegwee, R. A., & Spil, T. A. (2015). The six p's of the next step in electronic patient records in the netherlands. *Health Policy and Technology*, *4*(2), 137–143.

Myles, A. J., Feudale, R. N., Liu, Y., Woody, N. A., & Brown, S. D. (2004). An introduction to decision tree modeling. *Journal of Chemometrics: A Journal of the Chemometrics Society*, *18*(6), 275–285.

Noh, N. B. M., & Basri, M. N. B. M. (2021). Phishing website detection using random forest and support vector machine: A comparison. In *2021 2nd international conference on artificial intelligence and data sciences (aidas)* (pp. 1–5).

NU.nl. (2022). *Ict-storing bij ziekenhuis maastricht voorbij: afspraken op vrijdag gaan door.* https://www.nu.nl/tech/6222754/ict-storing-bij-ziekenhuis-maastricht-voorbij-afspraken-op-vrijdag-gaan-door.html. (Accessed: 2022-09-22)

Oliphant, T. E. (2006). *A guide to numpy* (Vol. 1). Trelgol Publishing USA.

Omar, S., Ngadi, A., & Jebur, H. H. (2013). Machine learning techniques for anomaly detection: an overview. *International Journal of Computer Applications*, *79*(2).

Pachauri, G., & Sharma, S. (2015). Anomaly detection in medical wireless sensor networks using machine learning algorithms. *Procedia Computer Science*, *70*, 325–333.

Pajouh, H. H., Dastghaibyfard, G., & Hashemi, S. (2017). Two-tier network anomaly detection model: a machine learning approach. *Journal of Intelligent Information Systems*, *48*(1), 61–74.

Powers, D. M. W. (2011). Evaluation: From precision, recall and f-measure to roc, informedness, markedness correlation. *Journal of Machine Learning Technologies*, *2*(1), 37-63.

Priestman, W., Sridharan, S., Vigne, H., Collins, R., Seamer, L., & Sebire, N. J. (2018). What to expect from electronic patient record system implementation: lessons learned from published evidence. *Journal of Innovation in Health Informatics*, *25*(2), 92–104.

Probst, P., & Boulesteix, A.-L. (2017). To tune or not to tune the number of trees in random forest. *The Journal of Machine Learning Research*, *18*(1), 6673–6690.

Probst, P., Wright, M. N., & Boulesteix, A.-L. (2019). Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: data mining and knowledge discovery*, *9*(3), e1301.

Ray, S., McEvoy, D. S., Aaron, S., Hickman, T.-T., & Wright, A. (2018). Using statistical anomaly detection models to find clinical decision support malfunctions. *Journal of the American Medical Informatics Association*, *25*(7), 862–871.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural computation*, *13*(7), 1443–1471.

Sharma, S., Gosain, A., & Jain, S. (2022). A review of the oversampling techniques in class imbalance problem. In *International conference on innovative computing and communications* (pp. 459–472).

Shelke, M. S., Deshmukh, P. R., & Shandilya, V. K. (2017). A review on imbalanced data handling using undersampling and oversampling technique. *Int. J. Recent Trends Eng. Res*, *3*(4), 444–449.

Shreyas, R., Akshata, D., Mahanand, B., Shagun, B., & Abhishek, C. (2016). Predicting popularity of online articles using random forest regression. In *2016 second international conference on cognitive computing and information processing (ccip)* (pp. 1–5).

Sittig, D. F., Lakhani, P., & Singh, H. (2022). Applying requisite imagination to safeguard electronic health record transitions. *Journal of the American Medical Informatics Association*, *29*(5), 1014–1018.

Susmaga, R. (2004). Confusion matrix visualization. In *Intelligent information processing and web mining* (pp. 107–116). Springer.

Suthaharan, S. (2016). Support vector machine. In *Machine learning models and algorithms for big data classification* (pp. 207–235). Springer.

Tan, X., Su, S., Huang, Z., Guo, X., Zuo, Z., Sun, X., & Li, L. (2019). Wireless sensor networks intrusion detection based on smote and the random

forest algorithm. *Sensors*, *19*(1), 203.

Tesfahun, A., & Bhaskari, D. L. (2013). Intrusion detection using random forests classifier with smote and feature reduction. In *2013 international conference on cloud & ubiquitous computing & emerging technologies* (pp. 127–132).

Timčenko, V., & Gajin, S. (2018). Machine learning based network anomaly detection for iot environments. In *Icist-2018 conference* (pp. 196–201).

Tsai, C.-F., & Lin, W.-C. (2021). Feature selection and ensemble learning techniques in one-class classifiers: an empirical study of two-class imbalanced datasets. *IEEE Access*, *9*, 13717–13726.

van der Graaf, P. (2012). *Epr in the dutch hospitals - a decade of changes: a study about epr system's success factors in the dutch hospitals* (Unpublished master's thesis). University of Twente.

van Veldhuizen, A. (2022). *Storing ziekenhuis zwolle na uren opgelost; operatiekamers en hartlonghulp weer open.* https://www.ad.nl/zwolle/storing-ziekenhuis-zwolle-na-uren-opgelost-operatiekamers-en-hartlonghulp-weer-open~a6274b42/. (Accessed: 2022-09-22)

Wang, S., Dai, Y., Shen, J., & Xuan, J. (2021). Research on expansion and classification of imbalanced data based on smote algorithm. *Scientific Reports*, *11*(1), 1–11.

Wilman, S. (2022). 'de basis is vertrouwen'. *Zorgvisie tech*, *23*(1), 30–32.

Wu, T., Fan, H., Zhu, H., You, C., Zhou, H., & Huang, X. (2022). Intrusion detection system combined enhanced random forest with smote algorithm. *EURASIP Journal on Advances in Signal Processing*, *2022*(1), 1–20.

Xing, H.-J., & Ji, M. (2018). Robust one-class support vector machine with rescaled hinge loss function. *Pattern Recognition*, *84*, 152–164.

Yeng, P. K., Fauzi, M. A., & Yang, B. (2020). Workflow-based anomaly detection using machine learning on electronic health records' logs: A comparative study. In *2020 international conference on computational science and computational intelligence (csci)* (pp. 753–760).

Yu, T., & Zhu, H. (2020). Hyper-parameter optimization: A review of algorithms and applications. *arXiv preprint arXiv:2003.05689*.

Zwetsloot-Schonk, J. H. M. (2003). De wonderlijke wereld van ict in de zorg.

APPENDIX A

For linear separable data, the kernel function would be as seen in Equation 12. This function calculates the dot product of two datapoints.

$$K(x_i, x_j) = x_i^T x_j \tag{12}$$

The most common kernel function is Radial Basis Function, seen in Equation 13. This function computes the similarity between two datapoints.

$$K(x_i, x_j) = exp(-\gamma||x_i - x_j||)^2, \gamma > 0 \tag{13}$$

This function calculates the squared euclidean distance between two samples.

Another kernel function is the polynomial function, seen in Equation 14.

$$K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0 \tag{14}$$

d is the polynomial degree used, r is the independent coefficient term used in the function.

Another kernel function is the sigmoid function, seen in Equation 15. This function uses the dot product of the vectors of x and y, and uses an independent coefficient term.

$$K(x_i, x_j) = tanh(\gamma x_i^T x_j + r) \tag{15}$$

The output is put through the hyperbolic tangent function.

Python (version 3.8.11) is used as the programming language in this thesis. The code is written in Jupyter Notebooks (Kluyver et al., 2016). The code was executed on a personal computer with an Intel Core i5-7300HQ CPU.

Table B1 provides an overview of the used libraries, their version and a short description.

Table B1: Overview of used libraries, including version, in alphabetical order.

Library	Description
Imbalanced-learn (0.6.0) (Lemaître, Nogueira, & Aridas, 2017)	Used for applying Synthetic Minority Oversampling to the dataset
Numpy (1.20.3) (Oliphant, 2006)	Used for working with arrays.
Pandas (1.3.3) (McKinney, 2011)	Used for processing the dataset
Seaborn (0.11.2) (Bisong, 2019)	Used to visualise the pairplot and confusion matrices.
Scikit-learn (1.0.0) (Hao & Ho, 2019)	Used to split the data into train, validation and test sets. Used to binarise labels and standardise data. Used for designing and applying all models on the data. This library was also used to calculate the recall, precision and F_1 scores and to generate the confusion matrices.

APPENDIX C

Table C1: Descriptive statistics for anomalies in the dataset used for SMOTE, rounded at two decimals. Time in seconds.

	Condition			
	Total Time	Query Time	Query Count	Query Row Count
Mean	11.83	8.09	1756.19	18608.77
Std	17.41	14.65	1758.57	34150.13
Min	0.56	0.00	0.00	0.00
25%	6.83	2.26	539.00	3692.00
50%	8.96	5.55	1676.00	6404.00
75%	10.72	7.16	2210.00	24382.00
Max	1081.97	612.66	85878.00	2434774.00

Table D1: F_1 score of each hyperparameter combination of the Support Vector Machine evaluated on the validation set. Highest F_1 score in bold. Note: the linear kernel function does not use the gamma hyperparameter.

	Gamma											
	0.1				0.01				0.001			
	C				C				C			
Kernel	0.1	1	10	100	0.1	1	10	100	0.1	1	10	100
Polynomial	0.969	0.982	0.983	0.983	0.828	0.892	0.949	0.969	0.099	0.334	0.501	0.828
Radial Basis Function	0.969	0.983	0.989	**0.991**	0.908	0.947	0.967	0.978	0.853	0.889	0.915	0.947
Sigmoid	0.068	0.068	0.068	0.068	0.733	0.724	0.723	0.722	0.842	0.870	0.857	0.856
Linear	0.896	0.897	0.897	0.897	0.896	0.897	0.897	0.897	0.896	0.897	0.897	0.897

Table D2: F_1 score, rounded at 4 decimals, of each hyperparameter combination of the Random Forest evaluated on the validation set. Highest F_1 score in bold.

	Number of Features	
Number of Trees	2	4
100	0.9998	0.9998
200	0.9998	**1.000**
300	0.9998	0.9998
400	0.9998	0.9998
500	0.9998	0.9998

Table D3: F_1 score of each hyperparameter combination of the One-Class Support Vector Machine evaluated on the validation set. Highest F_1 score in bold. Note: the linear kernel function does not use the gamma hyperparameter.

	Kernel											
	Polynomial			Radial Basis Function			Sigmoid			Linear		
	Gamma			Gamma			Gamma			Gamma		
v	0.1	0.01	0.001	0.1	0.01	0.001	0.1	0.01	0.001	0.1	0.01	0.001
0.01	0.000	0.000	0.000	0.273	0.276	**0.872**	0.208	0.060	0.574	0.004	0.004	0.004
0.03	0.000	0.015	0.000	0.252	0.716	0.721	0.126	0.650	0.518	0.026	0.026	0.026
0.05	0.000	0.001	0.000	0.238	0.624	0.630	0.420	0.624	0.036	0.662	0.662	0.662
0.07	0.000	0.000	0.000	0.226	0.565	0.570	0.394	0.333	0.018	0.389	0.389	0.389
0.09	0.007	0.000	0.000	0.473	0.519	0.525	0.346	0.090	0.041	0.122	0.122	0.122

Table D4: F_1 score, rounded at 4 decimals, of each hyperparameter combination of the Isolation Forest evaluated on the validation set. Highest F_1 score in bold.

	Number of Trees				
Sub-Sample Size	100	200	300	400	500
256	0.249	0.269	0.255	0.246	0.259
512	0.305	0.297	0.292	0.288	0.292
1024	0.347	0.330	0.342	0.337	0.335
2048	0.374	0.363	0.362	0.366	0.369
4096	0.411	0.373	0.387	0.385	0.388
8192	0.388	0.406	0.413	0.407	0.406
16384	0.474	0.451	0.447	0.446	0.441
32768	0.474	0.455	0.460	0.460	0.467
65536	0.500	0.492	0.485	0.488	0.489
131072	0.490	0.508	0.501	0.496	0.498
262144	0.530	0.530	0.520	0.522	0.517
411559	**0.539**	0.532	0.526	0.524	0.531

Table E1: False positives classified by the Support Vector Machine. Time in seconds, rounded at three decimals.

Duration	Total Query Duration	Query Count	Total Query Row Count
14.100	5.415	1473	19809
8.374	5.944	1997	3706
8.346	5.995	1986	3694
8.351	5.981	1997	3716
7.684	5.642	1937	7423
8.358	6.000	1997	3728
6.634	5.975	180	261
8.309	5.920	1991	3696
7.842	5.994	773	4361
6.412	5.464	885	18737

Table E2: First 50 false positives classified by the Synthetic Minority Over-Sampling Method-Support Vector Machine. Time in seconds, rounded at three decimals.

Duration	Total Query Duration	Query Count	Total Query Row Count
2.291	1.233	92	19623
5.638	1.223	756	19550
11.578	5.244	797	3112
1.557	0.536	257	19686
11.237	5.484	778	2909
8.111	1.960	1113	18838
6.499	2.102	1263	19402
8.216	5.829	1988	3687
2.188	1.601	422	19801
2.477	1.215	89	19660
11.692	5.285	764	2866
10.605	4.980	1967	6804
7.846	5.469	1991	3696
8.121	2.081	1695	18271
6.251	2.975	1955	3638
5.075	1.786	512	19621
4.898	1.093	524	19511
14.842	0.061	31	46
4.036	1.688	1031	19652
3.885	1.087	872	19957
7.791	5.424	1982	3685
4.550	2.245	1295	18705
4.481	3.656	82	18622
10.41	5.415	1473	19809
5.359	1.538	1336	19833
8.601	5.465	1789	3909
4.134	1.225	882	19962
7.890	5.465	1997	3710
7.974	5.627	1991	3696
8.374	5.944	1997	3706
11.478	5.482	798	3152
3.965	1.378	1023	19767
2.164	1.204	82	19680
7.986	5.625	1917	3635
8.017	5.566	1991	3698
7.851	5.482	1988	3687
5.166	2.808	1966	3152
8.346	5.995	1986	3694
4.547	1.642	812	19339
7.840	5.440	1991	3694
4.157	1.470	985	19814
7.288	2.508	1901	17880
6.234	5.846	194	212
6.451	3.745	1760	18979
7.774	5.394	1986	3687
8.049	5.643	1997	3710
1.993	1.485	222	19988
8.196	5.629	1991	3700

Table E3: False positives classified by the Synthetic Minority Over-Sampling Method-Random Forest. Time in seconds, rounded at three decimals.

Duration	Total Query Duration	Query Count	Total Query Row Count
10.410	5.415	1473	19809
14.896	1.162	720	3271
8.722	5.891	1013	2404

Table E4: First 50 false positives classified by the One-Class Support Vector Machine. Time in seconds, rounded at three decimals.

Duration	Total Query Duration	Query Count	Total Query Row Count
3.375	1.898	1677	6816
7.486	5.546	43	54
9.143	2.515	1880	8949
2.291	1.233	92	19623
3.260	0.461	205	19335
2.908	1.311	62	17975
11.578	5.244	797	3112
1.527	0.91	38	15706
1.557	0.536	257	19686
9.709	4.768	1850	6324
3.620	1.848	1713	2456
11.982	0.021	10	8
11.237	5.484	778	2909
5.807	2.121	1651	6900
8.872	4.369	1245	12589
1.229	0.888	35	15651
0.482	0.132	42	13867
8.111	1.960	1113	18838
13.602	0.057	28	60
6.848	5.428	60	99
2.181	1.241	60	18211
6.499	2.102	1263	19402
8.216	5.829	1988	3687
2.477	1.215	89	19660
6.683	2.305	1935	7309
11.692	5.285	764	2866
2.842	1.370	120	18151
10.605	4.98	1967	6804
1.263	0.928	34	15669
7.846	5.469	1991	3696
0.547	0.146	6	15041
11.29	2.473	1468	5710
3.694	1.778	1836	4495
8.121	2.081	1695	18271
9.121	0.038	16	11
9.024	2.742	1374	11703
6.375	2.399	1572	13757
2.088	1.198	74	18008
2.306	0.000	109	19016
7.692	2.181	1463	14159
3.736	2.700	1694	2966
1.465	1.164	64	17936
6.251	2.975	1955	3638
4.419	2.003	1900	4983
1.151	0.346	61	16006
7.376	3.854	1260	16012
14.842	0.000	31	46
1.522	0.492	329	18725
1.443	1.195	128	18150
8.509	4.265	1924	7001

Table E5: First 50 false positives classified by the Isolation Forest. Time in seconds, rounded at three decimals.

Duration	Total Query Duration	Query Count	Total Query Row Count
8.386	1.601	1184	4667
5.412	1.64	1353	15383
3.375	1.898	1677	6816
3.632	1.433	1554	10038
0.979	0.174	173	8050
5.251	3.013	798	10241
2.33	0.964	571	13785
9.997	4.957	681	3064
4.403	1.303	990	4394
9.125	1.491	1086	912
7.486	5.546	43	54
9.143	2.515	1880	8949
2.723	1.074	461	13388
0.955	0.622	470	5502
0.968	0.337	158	8693
2.174	0.996	617	8396
2.291	1.233	92	19623
6.203	0.752	274	7058
7.715	1.121	330	7565
7.588	1.283	648	18813
5.686	1.461	449	7149
7.164	2.558	345	7519
3.26	0.461	205	19335
4.565	2.875	531	3571
2.908	1.311	62	17975
6.792	1.422	830	4007
4.928	3.545	220	3354
3.538	2.882	174	3049
5.638	1.223	756	19550
2.682	1	530	8381
11.578	5.244	797	3112
0.922	0.579	488	5514
7.663	1.354	491	10901
5.201	2.97	1477	2813
6.764	4.366	712	3336
6.071	1.661	764	4361
1.567	0.171	58	10952
3.478	2.674	978	2556
1.557	0.536	257	19686
2.625	1.382	1091	3653
6.984	2.089	1215	7132
4.411	1.497	1393	6362
5.32	3.077	583	4791
9.947	1.033	603	2731
0.966	0.328	139	9227
6.252	1.333	726	8224
9.709	4.768	1850	6324
3.237	1.481	1235	1697
3.62	1.848	1713	2456
11.982	2.01	10	8