

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Scuola di Ingegneria e Architettura

Corso di Laurea Magistrale in INGEGNERIA INFORMATICA

Tesi di Laurea Magistrale in PROCESSI E TECNICHE DI DATA MINING M

Sistemi per la rilevazione di frodi nel “Carrier Billing”

Candidato:
Davide Bellettini

Relatore:
Chiar.mo Prof. Ing Claudio Sartori

Correlatori:
Dott.sa Francesca Cappannini
Dott. Mirko Bonadei

Anno Accademico 2014/2015 - Sessione III

Abstract

Il rilevamento e l'analisi delle frodi è una delle attività cruciali per un sistema di *carrier billing*. Una frode non rilevata prontamente può causare ingenti danni economici, mentre un falso positivo porta ad uno spreco di tempo da parte del team di *Reporting and Control*. In questa tesi viene studiato il dominio di un Payment Service Provider (PSP) operativo nel settore del carrier billing, andando ad approfondire il sistema di rilevamento delle frodi basato sull'analisi di serie storiche con la tecnica Holt-Winters. Verrà fornita una panoramica sull'architettura del sistema seguita da alcuni esempi in cui la scarsa qualità delle predizioni ha causato una perdita economica o temporale al PSP. Verranno quindi proposte numerose soluzioni per estendere e migliorare il sistema attuale, concentrandosi principalmente sulla pulizia dei dati da analizzare e sullo sfruttamento di informazioni implicitamente contenute nei dati. I miglioramenti apportati possono essere divisi in due categorie: quelli che necessitano della supervisione dell'essere umano e quelli che possono essere ottenuti in modo automatico da un algoritmo. Alcune di queste soluzioni verranno implementate e commentate confrontando le prestazioni del sistema prima e dopo le varie implementazioni. Alcune proposte verranno invece trattate solamente a livello teorico ma faranno parte degli sviluppi futuri. Infine si cercherà di trarre delle conclusioni, dimostrando come nel dominio del carrier billing sia possibile ottenere prestazioni soddisfacenti grazie ad un filtraggio supervisionato dei dati delle serie storiche, mentre i tentativi di filtraggio non supervisionato hanno fornito risultati contrastanti.

Parole chiave: data mining, serie storiche, rilevazione delle frodi, carrier

billing, Holt-Winters.

Ringraziamenti

Finalmente questa carriera universitaria volge al termine e lo devo anche a voi, che ho incrociato durante questo faticoso cammino!

Ringrazio il mio relatore Prof. Sartori, i miei correlatori aziendali Francesca Cappannini e Mirko Bonadei (che mi ha supportato e sopportato dalla scrivania a fianco alla mia) tutto le persone con cui ho lavorato in Onebip (nel presente e nel passato) a cominciare da Francesco Dominidiato, Giorgio, Danilo, Marco, Federico (x 2), Jacopo (x 2), Luca, Francesca A., Demo, Sabina, Carmen, RacaNet, Marco, Simone, Aurélien, Karim, Fabio, Andrea, Giulia, Rosario, Era, Paola, Antonella, Manu (x 3), Gabriele, Gokmen, Can, Francesco Tassi, Giuseppe, Filippo, Gigi, Gianluca.

Grazie a mio fratello ai miei cugini tra cui Romina e Piergiorgio, i miei zii e zie, la nonna 'Talina e tutti quanti, ma soprattutto grazie mamma, per tutto!

Grazie a Rossella e Veronica per l'ospitalità nei miei primissimi mesi milanesi, e ovviamente grazie a JJ Nardiello e visto che siamo in tema Lombardia: Sara, Natalino, Manu, Mara, Jonathan, Nathan.

Un abbraccio agli amici e le amiche del Saloon, che con me condividono il desiderio di capire perché queste stelle non si spengono (solo per darci un po' di complicità), e ovviamente Folco Orselli per aver scritto quella canzone meravigliosa e tutti gli altri capolavori dei suoi cinque dischi. Un caloroso saluto a tutti gli amici soci e simpatizzanti dell'associazione CPI.

Non sarei lo stesso senza gli amici della Romagna (Romagna mia lontan da te non si può star!), Davide e Daniele V., Giulia, Ilaria, Alex Must, Nicoletta, Paolo S., Raffo, Stefano, Zakk, Jack Bart, Alex, Gian Marco, Paolo C., i Di

Berardino, Ms. Freeza, Gabry, Biacco, Nicolò, Raniero, Cristiana, Elena.

Andando un po' più a nord ma non troppo un grazie agli amici di Univer-siBO e di Bologna in generale: Filippo, Fafà, Ilias, Andrea, Matteo, Miguel, Dece, Campa, Eddy, sissi, spiff, Emanuele, Gilbert, Adelmo, Alessio e Ghe-do, il Prof. Cristiano Verondini, lo zio Ferri, Link, countz, Abus, Magda, Martino, Michele, Lorenzo, Giuseppe, Franz Flor, Monta, Pablo, Francesco Morelli, Gero, Federico Stanzani, TelespallaFede, VideoQuerz, la Ceci, gli amici del pugBO: Tassoman, Bonzo, Francesco, Hoang e Yoghi.

Ringrazio Domenico Loparco per avermi aiutato a migliorare la mia tec-nica con il basso elettrico, i Never Too Late, i Beoir, gli Zampieri85, “Gli Atroci”, le Pecore Grigie, gli Shiny Madness, gli Shameless Dreams, i Cattivo Tempo, e tutti i gruppi musicali presenti e passati dei miei amici per aver trascorso insieme tanti momenti divertenti facendo musica o ascoltandola rigorosamente dalla prima fila sotto il palco.

Infine credo sia giusto ricordare chi avrebbe partecipato con gioia alla mia laurea ma che di recente ci ha lasciato, come mia nonna Giovanna e mio zio Adriano, siete vivi e presenti nei miei ricordi!

Per l'impaginazione di questa tesi, come per la triennale, è stato utilizzato L^AT_EX. Ringrazio Ilias Bartolini per aver reso pubblico il sorgente della propria tesi su Univer-siBO, agevolando notevolmente il mio lavoro.

Ringrazio anche chi ho sbadatamente dimenticato nella lista e meritava di esserci, posso sempre rimediare nella tesi di dottorato (scherzo, ma mai dire mai)!

Indice

Abstract	III
Ringraziamenti	V
Introduzione	XI
1 Il Carrier Billing	1
1.1 Mercato	1
1.1.1 Diffusione	2
1.1.2 Categorie merceologiche	2
1.1.3 Potere contrattuale	2
1.2 Flussi di pagamento	3
1.2.1 PIN	6
1.2.2 Operator	6
1.2.3 MO-MTB	6
1.2.4 MOB-MT	7
1.3 Le frodi	8
1.3.1 Definizione	8
1.3.2 Caratteristiche	9
1.3.3 Fattori di rischio	9
Cash flow	9
Chargeback	10
Spending limit	10
1.3.4 Case study	11
Conversione di credito	11

	Mancato addebito	11
1.4	L'azienda	12
1.5	Conclusioni	12
2	Sistema preesistente	13
2.1	Architettura	14
2.1.1	Tecnologie utilizzate	16
2.2	Serie storiche	16
2.2.1	Stagionalità	17
2.3	Il metodo Holt-Winters	18
2.3.1	Smorzamento esponenziale semplice	18
2.3.2	Smorzamento esponenziale doppio	18
	Valori iniziali	19
2.3.3	Smorzamento esponenziale triplo	19
	Modello con stagionalità moltiplicativa	19
	Modello con stagionalità additiva	20
2.3.4	Implementazione in R	21
	La procedura predefinita HoltWinters	22
2.4	Modello dei dati transazionali	23
2.5	Descrizione del processo antifrode	23
2.5.1	Controllo automatizzato	24
2.5.2	Controlli di routine	24
	Deleted Users	25
	Deleted MSISDNs	25
	Transfer-in vs. Payment-out	25
2.5.3	Analisi e azione	25
2.6	Esempi reali	26
2.6.1	Struttura dei grafici	26
	Serie storica e predizione	27
	Errore di predizione	27
	Parametri dell'algoritmo	27
2.6.2	Veri positivi	27
	Caso 1	28

Caso 2	30
2.6.3 Falsi positivi	31
Caso 1	31
Caso 2	33
2.6.4 Falsi negativi	33
Caso 1	34
Caso 2	34
2.7 Criticità	34
2.8 Conclusioni	37
3 Possibili miglioramenti del sistema esistente	39
3.1 Metriche di benchmark	39
3.2 Lunghezza del periodo di training	40
3.2.1 Falso positivo, caso 1	41
Parametri di smorzamento	41
MSPE della fase di training	46
3.2.2 MSPE della fase di predizione	47
3.3 Ulteriori dimensioni di aggregazione	48
3.3.1 Business model	48
3.3.2 Credit provider	51
3.4 Correzione supervisionata dei dati	52
3.5 Miglioramento della user experience	54
3.5.1 Granularità variabile	54
3.5.2 Gestione dei nuovi merchant	55
3.5.3 Livelli graduali di allarme	55
3.6 Holt-Winters Robusto	56
3.6.1 Confronto con la versione tradizionale	60
3.6.2 Implementazione	61
3.7 Conclusioni	62
4 Sviluppi futuri	65
4.1 Componenti multiple di stagionalità	65
4.1.1 Stagionalità doppia	66

4.1.2	Modello a stagionalità multipla	69
4.2	Conclusioni	72
A	Codice sorgente	73
A.1	Predizione mediante Holt-Winters	73
A.2	Procedura HoltWinters standard di R	75
A.3	Predizione mediante Supervised Holt-Winters	75
A.4	Procedura Supervised Holt-Winters	76
A.4.1	Parte R	76
A.4.2	Parte C++	80
A.5	Predizione mediante Robust Holt-Winters	82
A.6	Procedura Robust Holt-Winters	84
A.6.1	Parte R	84
A.6.2	Parte C++	87
	Conclusioni	95
	Glossario	97
	Bibliografia	101

Introduzione

Il *carrier billing*, ovvero il pagamento attraverso gli operatori telefonici utilizzando il credito del telefono cellulare, è una realtà piuttosto vasta ed in espansione, in particolare nei paesi dell'America latina. Contestualmente alla crescita dei volumi di affari la problematica della gestione delle frodi diventa sempre più rilevante.

Lo scopo di questa tesi è quello di analizzare le caratteristiche del sistema per la rivelazione delle frodi del Payment Service Provider (PSP) Onebip S.p.A. L'azienda si occupa di erogare un servizio di pagamento in carrier billing, basato su accordi commerciali che hanno permesso di implementare integrazioni con gli operatori telefonici al fine di fornire una soluzione di pagamento unica ai fornitori di beni e servizi digitali.

Nel capitolo 1 andremo ad illustrare le caratteristiche del mercato del carrier billing, la diffusione, il potere contrattuale delle parti in gioco ed il funzionamento dei principali flussi di pagamento, concludendo con un approfondimento sulle frodi.

Nel capitolo 2 mostreremo l'architettura generale del PSP, passando poi ad un'analisi dettagliata del processo antifrode, il quale si avvale dell'operatività degli esperti assistiti da un sistema automatico basato sull'analisi delle serie storiche con la tecnica Holt-Winters. Illustreremo esempi di vero positivo, falso positivo e falso negativo con grafici e commenti sulle caratteristiche dei singoli casi.

Nel capitolo 3 andremo a migliorare il sistema esistente, dapprima mantenendo l'algoritmo ma cambiando le caratteristiche dei dati utilizzati, aumentando la lunghezza del periodo di training e scegliendo diversi partiziona-

menti dei dati. Andremo poi ad implementare una correzione supervisionata dei dati e successivamente la versione robusta dell'algoritmo [GFC10].

Infine nel capitolo 4 andremo ad analizzare dal punto di vista teorico i possibili sviluppi futuri, come ad esempio modellazione delle stagionalità multiple.

Capitolo 1

Il Carrier Billing

In questo capitolo andremo ad introdurre nel dettaglio le caratteristiche e le dinamiche del mondo del del *carrier billing*, specialmente dal punto di vista del *PSP*; illustreremo le dinamiche di mercato, descrivendo il modello dei flussi di pagamento ed infine faremo una approfondita panoramica sulle caratteristiche delle frodi più comuni del settore e nei confronti delle quali è necessario adottare meccanismi di prevenzione e difesa.

1.1 Mercato

Il carrier billing rappresenta un'ampia fetta di un mercato ancora più vasto che è quello del *mobile payment*; in quest'ultima definizione rientrano tutte le tipologie di transazioni monetarie che coinvolgono, completamente o in parte l'utilizzo di un dispositivo mobile. L'aspetto caratteristico del carrier billing è il coinvolgimento dell'operatore telefonico dal punto di vista finanziario, e viene denominato per questo *credit provider*. Il denaro infatti proviene dal credito telefonico dell'utente in caso di scheda prepagata o viene addebitato nella prima fattura utile in caso di contratto postpagato; l'utenza telefonica viene denominata *credit device*.

Il ruolo del PSP è quello di fare da intermediario tra gli esercenti (merchant) e i credit provider; talvolta il rapporto tra PSP e credit provider può essere a sua volta mediato da un'azienda intermedia detta aggregatore.

1.1.1 Diffusione

La diffusione così capillare del telefono cellulare permette di coprire quella parte della popolazione mondiale provvista di cellulare ma priva di una carta di credito (*unbanked*) o con una scarsa propensione all'utilizzo online di questo strumento; per queste ragioni il conversion rate è tipicamente più alto rispetto alle carte di credito. Un altro aspetto da prendere in considerazione è quello della customer base ovvero il numero di utenti raggiunti; le stime parlano di circa 5 miliardi di utenti mobili a fronte di 1,5 miliardi di carte di credito.

1.1.2 Categorie merceologiche

La maggior parte delle normative nazionali prevede che il carrier billing possa essere utilizzato per l'acquisto di **solì beni e servizi digitali**; oltre all'aspetto normativo occorre precisare che nella maggior parte dei casi le commissioni applicate su questa tipologia di transazione vanno dal 30 al 70% rendendo sconveniente dal punto di vista commerciale la vendita di beni fisici. Esistono tuttavia importanti eccezioni: in alcuni mercati, come la Turchia gli operatori prevedono commissioni distinte per l'acquisto di beni digitali o fisici, in modo da rendere profittevole la vendita di questi ultimi.

1.1.3 Potere contrattuale

La natura del mercato fa sì che i credit provider operino in regime di oligopolio, questo comporta naturalmente un fortissimo potere contrattuale da parte loro, lo stesso vale per gli aggregatori.

Questo potere fa sì che sia difficile opporsi al mancato pagamento di una transazione, anche qualora la responsabilità sia interamente imputabile al credit provider.

Si sono verificati episodi di mancato pagamento di transazioni ritenute fraudolente anche dopo che il credit provider ne aveva confermato in prima battuta la legittimità.

Qualora le cifre coinvolte siano ingenti, questo comportamento da parte del credit provider può portare il PSP ad intraprendere lunghe azioni legali nel tentativo di ottenere il pagamento delle transazioni che esso ha già corrisposto ai merchant.

Dall'altro lato i PSP operano in un regime di forte concorrenza, il che porta i merchant ad integrarsi con il miglior offerente. La metodologia utilizzata dai merchant per scegliere il PSP si chiama A/B testing, il quale permette al merchant di capire quale sia il PSP che offre il miglior conversion rate, ovvero il miglior introito a parità di numero di persone inviate sulla pagina di pagamento. Questa continua ricerca da parte dei merchant del miglior PSP porta di conseguenza ad una *guerra dei prezzi*.

1.2 Flussi di pagamento

Nello svolgimento di una transazione di addebito nel carrier billing sono coinvolti molteplici attori: l'acquirente (o buyer) che intende acquistare un bene (item) o effettuare l'iscrizione ad un servizio in abbonamento (subscription), partendo dal sito Web del fornitore di beni o servizi digitali (merchant) viene indirizzato sulla pagina di pagamento del payment service provider il quale attraverso la sua integrazione con il credit provider provvede a completare l'acquisto. A seconda delle scelte tecnologiche e commerciali effettuate in una determinata nazione l'addebito dell'importo sul credito telefonico può essere strettamente legato all'acquisto oppure l'utente può accumulare del credito da spendere anche in un secondo momento. Quando l'acquisto è completato con o senza successo, il merchant riceve una *notifica* di avvenuto o mancato pagamento.

La molteplicità delle tecnologie disponibili nell'ambito e la varietà delle normative nazionali e delle politiche dei credit provider fanno sì che il *workflow* del processo di pagamento richieda di essere implementato secondo alcune varianti; dato che queste differenze possono avere un grosso impatto sulle possibili modalità di truffa sarà necessario approfondire le caratteristiche delle principali famiglie di flussi: PIN, Operator, mo-mtb, mob-mt.



The screenshot shows the Onebip payment interface. At the top left is the Onebip logo with the tagline "easy • monetization". To the right of the logo, the price "€ 2,90 (iva incl.)" and the text "Test ONEBIP TEST MERCHANT" are displayed. Below this, a green circle with the number "1" indicates the first step: "Inserisci il numero di telefono con cui effettuare l'acquisto". A text input field contains the number "612 34 56 78". Below the input field is a checked checkbox with the text "Ho letto l'Informativa sulla Privacy". At the bottom of the form is a green button labeled "CONTINUA".

Figura 1.1: Inserimento del numero di telefono (walk in)



The screenshot shows the Onebip payment interface at the PIN entry step. At the top left is the Onebip logo with the tagline "easy • monetization". To the right of the logo, the price "€ 2,90 (iva incl.)" and the text "Test ONEBIP TEST MERCHANT" are displayed. Below this, a green circle with the number "1" indicates the first step: "Il numero con cui stai per effettuare l'acquisto è 612345678". A second green circle with the number "2" indicates the second step: "Introduci il PIN che hai ricevuto sul tuo cellulare". A text input field contains the example PIN "Es. 4275". To the right of the input field is a green circular icon with a white "1" inside. At the bottom of the form is a green button labeled "ACQUISTA".

Figura 1.2: Inserimento del pin (opt-in)



Figura 1.3: Istruzioni (flusso mob-mt o mo-mtb)



Figura 1.4: Messaggio di conferma

1.2.1 PIN

Questo tipo di flusso è basato sull'identificazione dell'utente tramite Personal Identification Number (PIN) i tipici passi sono:

1. l'utente *atterra* sulla pagina di pagamento;
2. l'utente inserisce il numero di telefono (walk in) (figura 1.1);
3. il PSP invia un PIN per verificare il possesso del numero di telefono;
4. l'utente inserisce il PIN (figura 1.2)
5. il PSP verifica il PIN;
6. in caso di successo viene completato il pagamento e visualizzato un messaggio di conferma (figura 1.4);
7. il PSP notifica il successo o fallimento della transazione al merchant;
8. la pagina di pagamento visualizza un messaggio di conferma (figura 1.4).

1.2.2 Operator

Il credit provider possiede una propria pagina di pagamento, l'operazione viene quindi completamente delegata a quest'ultimo il quale notificherà il PSP dell'avvenuta transazione o dell'annullamento della stessa, informazione che verrà inoltrata al merchant da parte del PSP.

1.2.3 MO-MTB

Questo tipo di flusso è basato sull'invio di un messaggio gratuito a cui segue la risposta da parte del PSP mediante premium SMS, nel dettaglio:

1. l'utente *atterra* sulla pagina di pagamento;
2. l'utente, seguendo le istruzioni (figura 1.3), invia un SMS gratuito ad un numero speciale detto short code contenente una parola chiave che identifica l'importo;

3. il credit provider inoltra il messaggio al PSP;
4. il sistema del PSP verifica che l'utente non abbia superato il limite di spesa e che non sia stato bloccato;
5. il PSP procede all'invio di un messaggio di ringraziamento che causerà l'addebito della transazione;
6. il credit provider invia un messaggio di delivery report al PSP;
7. il PSP notifica il successo o fallimento della transazione al merchant;
8. la pagina di pagamento visualizza un messaggio di conferma (figura 1.4).

1.2.4 MOB-MT

Questo tipo di flusso è simile al precedente, soprattutto dal punto di vista dell'utente, ma la differenza più importante è che il messaggio inviato allo shortcode innesca l'addebito, mentre quello di risposta è gratuito, nel dettaglio:

1. l'utente *atterra* sulla pagina di pagamento;
2. l'utente invia un messaggio a pagamento allo short code indicato nelle istruzioni (figura 1.3);
3. il credit provider provvede ad addebitare l'importo (unico per shortcode);
4. solo in caso di successo il credit provider inoltra il messaggio al PSP;
5. il PSP, opzionalmente, provvede all'invio di un messaggio di ringraziamento;
6. il PSP notifica il successo della transazione al merchant;
7. la pagina di pagamento visualizza un messaggio di conferma (figura 1.4).

In sostanza, rispetto al flusso precedente, prima ancora che il PSP riceva il messaggio il credit provider addebita il costo della transazione; per questo motivo l'unica cosa possibile è registrare la transazione ed accreditarla sul saldo dell'utente, senza nessuna possibilità di controllo. Questa mancanza di controllo fa sì che sia più facile perpetrare frodi con questo flusso.

1.3 Le frodi

In questa sezione andremo ad affrontare le frodi, partendo prima con la definizione dei concetti, per poi descriverne le caratteristiche ed infine elencheremo una serie di casi reali di frode avvenuti tra il 2014 e il 2015.

1.3.1 Definizione

Phua et al [PLSG10] definiscono frode come l'abuso del sistema di una azienda senza necessariamente incorrere in conseguenze legali dirette; inoltre categorizzano i truffatori in tre categorie:

- **comune**: il truffatore comune adotta un comportamento disonesto in modo casuale e/o sporadico, qualora gli si presenti l'opportunità;
- **criminale**: il truffatore criminale che agisce individualmente;
- **organizzato**: come il criminale ma non agisce individualmente ma in associazione a delinquere.

La maggior parte delle frodi nel carrier billing avvengono da parte della prima categoria.

In questo lavoro di tesi definiamo frode sia i comportamenti degli utenti che portano al mancato pagamento delle transazioni da parte dei credit provider, sia quelli tecnicamente leciti ma che nei fatti possono danneggiare il PSP dal punto di vista della sua reputazione o dei rapporti commerciali con i merchant.

1.3.2 Caratteristiche

Nella maggior parte dei casi l'obiettivo dei truffatori è quello di convertire del credito telefonico ottenuto in modo illecito in denaro liquido (riciclaggio), una parte rilevante delle tecniche fraudolente sfrutta le caratteristiche dei giochi di ruolo online (MMORPG):

- **compravendita di personaggi:** il truffatore crea un personaggio ed utilizza il credito telefonico per *potenziarne* le caratteristiche e gli *armamenti* per poi cederlo ad altri giocatori in cambio di denaro;
- **gold farming:** il truffatore acquista del denaro virtuale mediante credito telefonico per poi scambiarlo con denaro reale;
- **cessione di account:** un utente che vuole convertire il proprio credito telefonico in denaro *presta* il proprio account registrato presso il PSP per permettere a terzi l'acquisto.

1.3.3 Fattori di rischio

Nella gestione delle transazioni nel carrier billing occorre esaminare diversi fattori di rischio: andremo ad illustrare i flussi di cassa (*cash flow*), le difficoltà commerciali nell'effettuare degli storni (*chargeback*) e l'importanza dei limiti di spesa (*spending limit*).

Cash flow

I flussi di cassa rappresentano il fattore di rischio principale in quanto svantaggioso per il PSP, i credit provider confermano la legittimità delle transazioni con tempi molto lunghi, è frequente che l'invio dei *benestare per l'emissione della fattura* avvenga tra i 90 e i 180 giorni dalla fine del mese di riferimento. Fino a quel momento il PSP è all'oscuro del numero di transazioni effettivamente riconosciute. Il pagamento verso il merchant avviene tipicamente in 60 giorni, questo significa che per un periodo che va dai 30 ai 120 giorni il PSP anticipa del denaro che non è certo di ricevere.

Chargeback

Qualora l'ammancio di transazioni sia significativo, il PSP può decidere di applicare al merchant uno storno (chargeback) sui pagamenti successivi, la scelta deve essere valutata caso per caso in quanto:

- non è sempre possibile ricondurre le transazioni non pagate ad uno specifico merchant;
- per ragioni commerciali potrebbe essere sconveniente applicare un chargeback: vi è un costo di comunicazione con il merchant che potrebbe essere poco propenso ad accettare lo storno.

È sempre più frequente che i merchant richiedano un contratto di tipo *chargeback free*, ovvero a fronte di una commissione più elevata scaricano completamente il rischio dovuto alle frodi interamente sul PSP il quale da contratto non può effettuare operazioni di chargeback.

Spending limit

La gestione del limite della spesa mensile e giornaliero relativo al singolo buyer è rilevante e può essere effettuata secondo diverse politiche, in particolare può essere:

- stabilito e imposto dal credit provider tramite la sua infrastruttura tecnologica;
- stabilito dal credit provider dal punto di vista contrattuale, ma garantito dall'infrastruttura del PSP;
- non presente, ma stabilito dal PSP per precauzione.

Nel primo caso il credit provider è sollevato dalla gestione del limite, nel secondo caso, il mancato rispetto dello stesso da parte del PSP può essere ragione di mancato pagamento da parte del credit provider. Nel terzo caso si stabilisce arbitrariamente un limite plausibile oltre al quale è altamente probabile che si tratti di una truffa.

1.3.4 Case study

In questa sezione descriveremo alcuni episodi di frode realmente accaduti negli anni 2014 e 2015.

Conversione di credito

Nel 2014 una azienda si presentava agli utenti come partner del PSP (un chiaro tentativo di *phishing*) in grado di trasferire il saldo degli utenti verso PayPal in cambio di una commissione.

L'utente incrementava il proprio saldo presso il PSP e poi forniva le proprie credenziali al finto partner del PSP in modo che questo potesse utilizzare il credito per acquistare *item* presso i merchant, soprattutto MMORPG rivendendoli a prezzo scontato, creando una sorta di *mercato nero*. Questa operazione non danneggiava economicamente il PSP in termine di mancati pagamenti da parte del credit provider, ma creava un grosso danno di immagine e un forte rischio di perdere l'accordo commerciale con i merchant coinvolti, inoltre il modello facilitava la conversione di credito ottenuto fraudolentemente.

Mancato addebito

La modalità di frode più frequente è quella conseguente al mancato addebito, può infatti accadere che per un problema tecnico o bug le transazioni vadano a buon fine anche in assenza di credito; è frequente che gli utenti si accorgano di questo malfunzionamento e tentati dall'occasione inizino ad acquistare un grande quantitativo di beni digitali, alcuni esempi:

- gli utenti riuscivano ad inviare SMS premium senza pagare;
- il credit provider inviava notifiche con un codice di successo invece che di errore;
- un codice di errore del credit provider era stato mappato come codice di successo.

1.4 L'azienda

L'attività di tesi è stata svolta presso Onebip S.p.A. azienda di Milano che opera come PSP nell'ambito del carrier billing ed ISP billing.

1.5 Conclusioni

In questo capitolo abbiamo analizzato le caratteristiche del mercato e soprattutto gli aspetti che rendono il PSP un soggetto *debole* dal punto di vista del potere contrattuale, illustrati i flussi di pagamento ed analizzato le possibili frodi; la posizione commerciale del PSP rende fondamentale l'adozione di misure volte a prevenire e/o limitare i danni relativi alle frodi, per questo nel capitolo 2 andremo ad analizzare le misure che sono state adottate, sia dal punto di vista algoritmico sia come processo di business e riprenderemo gli episodi fraudolenti citati con esempi numerici.

Capitolo 2

Sistema preesistente

Lo scopo di questo capitolo è quello di descrivere lo stato dei sistemi, delle tecniche e dei processi di business messi in atto, precedentemente al nostro lavoro, al fine di prevenire o limitare i danni dovuti alle attività fraudolente degli utenti.

Per cominciare andremo ad illustrare l'architettura del sistema per poi affrontare la tematica delle *serie storiche*, prima in generale e poi con un approfondimento teorico sulla tecnica di smorzamento esponenziale Holt-Winters e la relativa implementazione in linguaggio R.

Un'importante parentesi sarà dedicata al modello dei dati transazionali, in quanto sorgente per le serie storiche analizzate.

Entrando nel vivo dell'argomento andremo ad analizzare in profondità gli elementi del processo di rilevamento e gestione dei comportamenti sospetti, sia nel suo aspetto automatico sia nell'operatività degli esperti del team di *reporting and control*.

Verranno esaminati alcuni episodi significativi accaduti durante l'operatività del PSP degli ultimi anni, con casi di *vero positivo*, *falso positivo* e *falso negativo*.

Infine, alla luce di quanto visto, esamineremo in maniera esaustiva le criticità dell'approccio adottato, in particolare per quanto riguarda la rapidità di intervento e la precisione (in termine di falsi positivi e negativi) delle tecniche adottate.

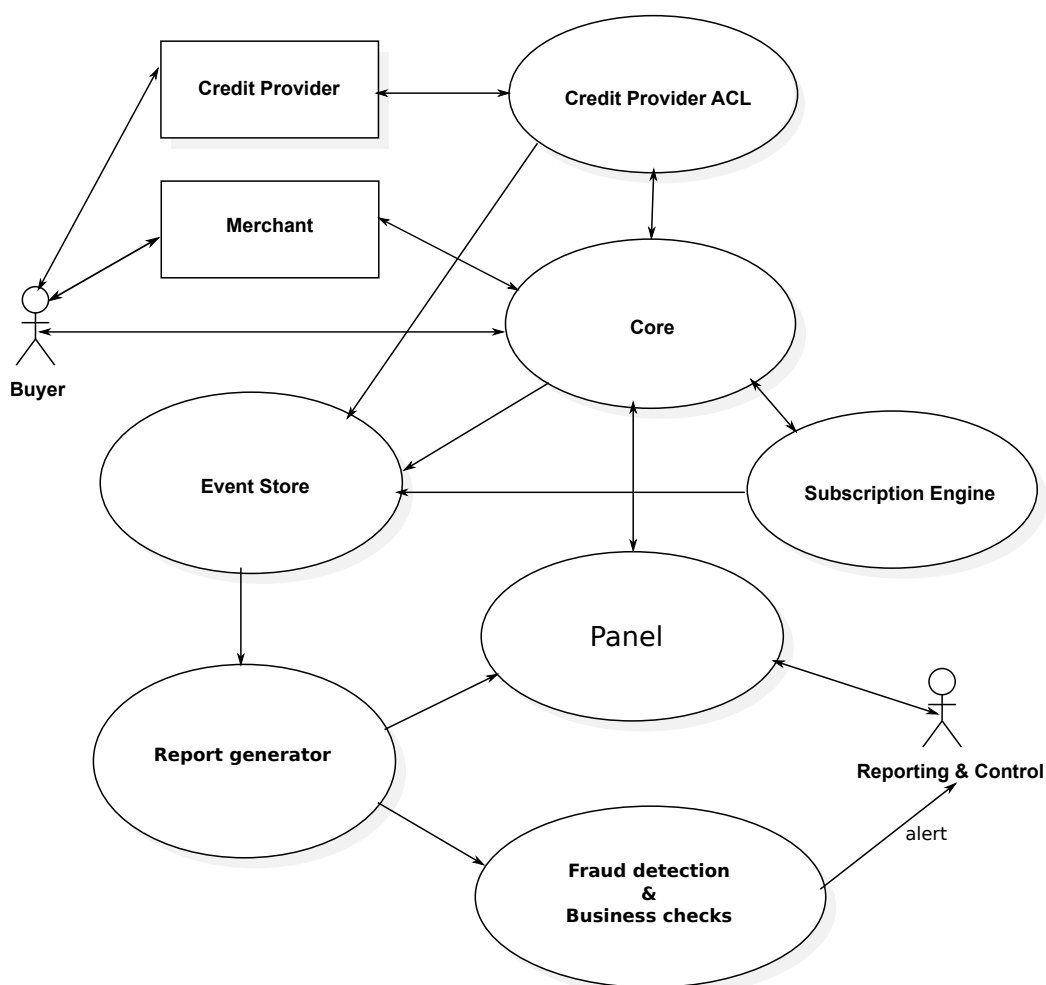


Figura 2.1: Architettura del sistema

2.1 Architettura

Per meglio comprendere l'operatività del sistema andremo a descrivere l'architettura dei sistemi di Onebip (fig. 2.1), con una particolare enfasi sulle interazioni con l'esterno (buyer, merchant e credit provider).

L'architettura è composta di servizi che comunicano tra di loro tramite Hypertext Transfer Protocol (HTTP) secondo lo stile REpresentational State Transfer (REST) [Fie00] in particolare abbiamo:

- il **sottosistema principale** (core): la parte centrale e per questo più

corposa del sistema, include le seguenti funzionalità:

- logica di business per il pagamento e contabilità;
 - pagina di pagamento;
 - integrazione con i sistemi dei merchant, per la gestione dei pagamenti e la notifica del risultato;
 - gestione degli utenti (operazioni di blocco/sblocco etc.)
- il **subscription engine**: è un motore, che interagisce esclusivamente con il sottosistema principale, per la gestione dei servizi in abbonamento, compito piuttosto complesso per via delle normative che regolano questo tipo di business; ad esempio: è necessario gestire il numero e la frequenza dei tentativi per l'attivazione o il rinnovo di un'iscrizione, le fasce orarie e i giorni della settimana in cui è possibile tentare un rinnovo, l'invio periodico di messaggi che ricordano agli utenti la propria iscrizione al servizio etc.;
 - il **credit provider anticorruption layer** [Eva04]: è un layer di astrazione sulle Application Programming Interface (API) dei credit provider ed ha il fine di evitare che le logiche necessarie all'integrazione, spesso piuttosto contorte, vadano a *corrompere* la qualità del codice del core;
 - l'**event store** [Ver13]: ha la responsabilità di raccogliere messaggi da tutti gli altri sottosistemi, questi messaggi seguono il *modello ad eventi* e rappresentano esplicitamente le transizioni di stato del sistema che sono di interesse per il business (es. pagamento completato, pagamento annullato etc.) e vengono messi a disposizione del sistema di reportistica;
 - il **sistema di reportistica**: a partire dagli eventi registrati nell'event store permette di generare report specializzati per i vari casi d'uso; una lista di esempi non esaustiva include i report *finanziari* (es. transfer e payment, cfr. 2.4), quelli per la fraud detection, quelli statistici (conversion rate, report di coorte), e quelli per il *troubleshooting*;

- il **pannello di controllo**: è lo strumento che visualizza le informazioni necessarie agli esperti del business per prendere decisioni operative (reportistica, informazione sugli utenti, lista delle transazioni etc.) e eseguire azioni sul sistema quali la gestione e blocco dei buyer o dei merchant, la gestione delle condizioni economiche riservate ai merchant etc.;
- il sistema per la **fraud detection** e controllo delle **metriche di business**: ha il compito di eseguire analisi sui dati generati dal sistema di reportistica, esistono varie tipologie di analisi che possono essere effettuate, che vanno dal controllare che una certa metrica non fuoriesca da un intervallo prestabilito ad analisi più approfondite mediante l'utilizzo di ambienti statistici; le analisi possono scatenare degli *alert* che possono essere inviati mediante sms o e-mail.

2.1.1 Tecnologie utilizzate

Per quanto riguarda l'insieme delle tecnologie, i sottosistemi sono stati sviluppati per il sistema operativo GNU/Linux, con i linguaggi PHP e JavaScript, utilizzando i Relational Database Management System (RDBMS) MySQL e PostgreSQL e il Database Management System (DBMS) non relazionale MongoDB; per quanto riguarda l'ambiente statistico la scelta è ricaduta sul linguaggio R.

2.2 Serie storiche

Per serie storica si intende una successione di valori legati a precisi istanti temporali, tipicamente equidistanti; il loro studio permette di effettuare analisi al fine di ottenere informazioni di interesse statistico. Alcuni possibili obiettivi nello studio delle serie storiche sono la previsione di valori futuri (*forecast*), la previsione dell'impatto di un evento e la ricerca di schemi causali tra variabili.

Le applicazioni di queste tecniche vanno dalla previsione delle vendite di una azienda al monitoraggio di indicatori economici o ambientali.

Per quanto riguarda le previsioni occorre precisare che i casi disponibili (passato) non sono mai rappresentativi della stessa popolazione oggetto della previsione, occorre quindi adottare un approccio differente; uno dei possibili approcci consiste nel supporre che la serie storica sia divisa in più componenti (di cui si andrà ad effettuare la stima) non rilevabili singolarmente:

- la **tendenza** (o trend) descrive l'andamento medio della serie può essere crescente, decrescente o stazionaria;
- la **stagionalità**: che descrive le fluttuazioni regolari e di breve periodo;
- la **fluttuazione casuale** o errore: la componente residua della serie storica, ottenuta dopo avere rimosso le componenti di tendenza e stagionalità, dovrebbe assumere le caratteristiche del *rumore bianco*.

Delle tre componenti illustrate, la stagionalità è quella che merita un approfondimento teorico, andremo quindi ora ad illustrare le sue caratteristiche e i possibili modelli (additivo e moltiplicativo).

2.2.1 Stagionalità

La stagionalità è la tendenza di una serie storica ad esibire comportamenti che si ripetono dopo s unità di tempo, esistono due tipi di stagionalità: quella additiva e quella moltiplicativa.

Le serie che presentano una stagionalità additiva hanno una componente, che si ripete periodicamente e non dipende dall'andamento generale (ad esempio le vendite di un determinato prodotto che aumentano di 1 milione di euro durante il mese di dicembre).

Nel caso della stagionalità moltiplicativa invece la componente periodica è relativa all'andamento generale (ad esempio un prodotto che aumenta le proprie vendite del 40% durante il mese di dicembre).

2.3 Il metodo Holt-Winters

Lo smorzamento esponenziale triplo, detto anche Holt-Winters (dai nomi degli autori) [Hol04, Win60, Kal04] è una tecnica per la previsione delle serie storiche; per una migliore chiarezza della descrizione introdurremo prima lo smorzamento esponenziale semplice, poi quello doppio e infine quello triplo.

2.3.1 Smorzamento esponenziale semplice

Lo smorzamento esponenziale semplice è una media pesata tutti gli elementi della serie, il peso decresce secondo una legge esponenziale, la serie smorzata verrà quindi calcolata secondo la formula:

$$\tilde{y}_t = \alpha y_t + (1 - \alpha)\tilde{y}_{t-1} \quad 0 \leq \alpha \leq 1 \quad (2.1)$$

Dove \tilde{y}_t è il valore t-esimo della serie smorzata e viene anche detto *livello*, α è il coefficiente di smorzamento e y_t è l'elemento t-esimo della serie storica originale. Il coefficiente di smorzamento regola quanto rapidamente il peso degli elementi *lontani* della serie decresce, fino ad avere comportamenti estremi nei casi limite $\alpha = 0$ e $\alpha = 1$. Con $\alpha = 0$ la serie derivante sarà costante e varrà \tilde{y}_1 , mentre con $\alpha = 1$ corrisponderà alla serie originale. Più il valore di α sarà piccolo, più la scelta del valore iniziale \tilde{y}_1 sarà determinante. Un possibile metodo di inizializzazione è la media aritmetica dei primi N valori (es. $N = 5$).

Questa tecnica assume che la serie storica oscilli attorno ad una media ragionevolmente stabile, il suo obiettivo è quindi quello di filtrare la fluttuazione casuale e può essere utilizzata per previsioni sul breve periodo.

2.3.2 Smorzamento esponenziale doppio

Lo smorzamento esponenziale doppio aggiunge una seconda componente in modo da tenere conto della tendenza:

$$\tilde{y}_t = \alpha y_t + (1 - \alpha)(\tilde{y}_{t-1} + F_{t-1}) \quad 0 < \alpha < 1 \quad (2.2)$$

$$F_t = \beta(\tilde{y}_t - \tilde{y}_{t-1}) + (1 - \beta)F_{t-1} \quad 0 < \beta < 1 \quad (2.3)$$

Oltre alla componente di livello \tilde{y}_t abbiamo F_t che è una stima smorzata della tendenza; β è il secondo coefficiente di smorzamento e y_t è il t-esimo elemento della serie storica originale.

Valori iniziali

Esistono più modi per scegliere i valori iniziali, \tilde{y}_1 viene tipicamente fissato a y_1 mentre alcune possibili scelte per F_1 sono:

$$F_1 = y_2 - y_1$$

$$F_1 = [(y_2 - y_1) + (y_3 - y_2) + (y_4 - y_3)]/3$$

$$F_1 = (y_n - y_1)/(n - 1)$$

2.3.3 Smorzamento esponenziale triplo

Lo smorzamento esponenziale triplo aggiunge un terzo elemento per tenere conto della stagionalità che, come già visto nella sezione (2.2.1), può avere una natura moltiplicativa o additiva; queste due nature danno vita a due distinti modelli Holt-Winters.

Modello con stagionalità moltiplicativa

Il modello esprime la serie storica secondo l'equazione:

$$y_t = (b_1 + b_2 t) S_t + \varepsilon_t \quad (2.4)$$

Dove

- b_1 è la *componente permanente*;
- b_2 è una componente lineare che rappresenta la tendenza;
- S_t è la componente di stagionalità moltiplicativa;
- ε_t è la fluttuazione casuale.

Sia s la periodicità stagionale i fattori S_t sono definiti in modo che valga sempre:

$$\sum_{1 \leq t \leq s} S_t = s \quad (2.5)$$

La componente di tendenza può essere eliminata dal modello se non necessaria. Il valore delle componenti viene stimato utilizzando le seguenti formule:

- La componente permanente:

$$a_t = \alpha \frac{y_t}{S_{t-s}} + (1 - \alpha)(a_{t-1} + a_{t-1}) \quad (2.6)$$

- La componente di tendenza:

$$F_t = \beta(a_t - a_{t-1}) + (1 - \beta)F_{t-1} \quad (2.7)$$

- La componente di stagionalità moltiplicativa:

$$S_t = \gamma \frac{y_t}{a_t} + (1 - \gamma)S_{t-s} \quad (2.8)$$

Con $\alpha, \beta, \gamma \in [0, 1]$. La previsione del valore della serie all'istante $t + h$ calcolata con gli elementi fino a t è quindi così definito:

$$\hat{y}_{t+h|t} = (a_t + hF_t)S_{t+h-qs} \quad q = \left\lceil \frac{h}{s} \right\rceil \quad (2.9)$$

Modello con stagionalità additiva

Il modello esprime la serie storica secondo l'equazione:

$$y_t = b_1 + b_2 t + S_t + \varepsilon_t \quad (2.10)$$

Dove

- b_1 è la *componente permanente*;
- b_2 è un componente lineare che rappresenta la tendenza;
- S_t è la componente di stagionalità additiva;
- ε_t è la fluttuazione casuale.

Sia s la periodicità stagionale i fattori S_t sono definiti in modo che valga sempre:

$$\sum_{1 \leq t \leq s} S_t = 0 \quad (2.11)$$

Il valore delle componenti viene stimato utilizzando le seguenti formule:

- La componente permanente:

$$a_t = \alpha(y_t - S_{t-s}) + (1 - \alpha)(a_{t-1} + F_{t-1}) \quad (2.12)$$

- La componente di tendenza:

$$F_t = \beta(a_t - a_{t-1}) + (1 - \beta)F_{t-1} \quad (2.13)$$

- La componente di stagionalità additiva:

$$S_t = \gamma(y_t - a_t) + (1 - \gamma)S_{t-s} \quad (2.14)$$

Con $\alpha, \beta, \gamma \in [0, 1]$. Il valore della previsione del valore all'istante $t + h$ è quindi così definito:

$$\hat{y}_{t+h|t} = a_t + hF_t + S_{t+h-qs} \quad q = \lceil \frac{h}{s} \rceil \quad (2.15)$$

2.3.4 Implementazione in R

L'implementazione in linguaggio R utilizzata (per il codice sorgente si veda A.1) è basata sulle librerie predefinite ed è per questo piuttosto semplice: accetta come parametro la periodicità e legge la serie storica in formato comma separated values (CSV) da *standard input*.

La serie storica ottenuta **escludendo l'ultimo valore** viene utilizzata per generare un modello di previsione con la procedura `HoltWinters` senza specificare alcun parametro [R C13]; dato che i valori di $(\alpha, \beta$ e $\gamma)$ non sono specificati la procedura li ricava autonomamente.

Qualora, come nel nostro caso, non venga impostato esplicitamente, il modello di stagionalità impiegato è quello additivo.

Il modello così ottenuto viene utilizzato, mediante la procedura `predict` per stimare il valore dell'ultimo elemento della serie storica (che era stato precedentemente escluso); la procedura restituisce un intervallo con livello di confidenza al 95%, il risultato dell'elaborazione viene scritto su *standard output*.

La procedura predefinita `HoltWinters`

Data la disponibilità del codice sorgente (si veda A.2) dell'ambiente R (sotto licenza GNU GPLv2) è possibile analizzare nel dettaglio il funzionamento *interno* della libreria: essa è composta da due parti, la parte in linguaggio C è una procedura che implementa l'algoritmo Holt-Winters vero e proprio, mentre la parte in R, di contorno, si occupa di:

- determinare i valori di partenza $a_1, F_1, S_{1..s}$ utilizzando le prime due stagioni (nel nostro caso settimane);
- eseguire il *binding* della procedura C in modo da permetterne l'invocazione;
- per determinare i valori ottimali per i parametri α, β e γ definisce il problema di minimizzazione utilizzando come metrica la somma dei quadrati degli errori di predizione, in inglese *sum of squared errors of prediction (SSE)* e lo risolve mediante l'algoritmo **s-BFGS-B** [ZBLN97] che esegue ripetutamente il training del modello Holt-Winters (escludendo i primi s elementi) variandone i parametri alla ricerca di un ottimo locale;
- restituire il modello con le componenti di livello, tendenza e stagionalità, oltre al valore di α, β e γ e SSE.

Una procedura secondaria `predict.HoltWinters` si occupa invece della predizione vera e propria, permette di prevedere uno o più valori, opzionalmente è possibile richiedere un intervallo di confidenza invece di un valore puntuale, il livello di confidenza può essere esplicitato, in caso contrario viene impostato al 95%.

2.4 Modello dei dati transazionali

La gestione dei flussi finanziari dal punto di vista del PSP prevede due tipologie di transazioni, *transfer* e *payment*, la prima rappresenta un trasferimento di denaro da o verso un'entità esterna, mentre i payment invece riguardano solamente flussi interni al sistema, entrambi possono essere in entrata (in) o in uscita (out), in particolare, in ordine logico:

1. il **transfer-in**: rappresenta un trasferimento di denaro dal credit provider al PSP, va ad incrementare il saldo del buyer;
2. il **payment-out**: rappresenta l'addebito di un pagamento sul saldo del buyer;
3. il **payment-in**: rappresenta l'accredito della quota del pagamento spettante al merchant;
4. il **transfer-out**: rappresenta la transazione bancaria di pagamento dal conto corrente del PSP a quello del merchant, che avviene normalmente con cadenza mensile.

La terminologia non ha carattere di generalità ma è specifica del payment service provider in cui è stata svolta l'attività di tesi; il modello è strettamente correlato al concetto di contabilità in *partita doppia*, sarà quindi **nella sostanza** analogo a quello di sistemi concorrenti.

Va inoltre introdotto, dato il suo utilizzo come metrica, il concetto di *gross transaction value (GTV)* ovvero l'importo pagato dall'utente finale per l'acquisto di un bene digitale o servizio. L'importo è espresso nella valuta originale del buyer ma viene anche convertito nella valuta del PSP, nel nostro caso l'Euro e in quella di riferimento del merchant (es. GBP, USD).

2.5 Descrizione del processo antifrode

In questa sezione andremo ad illustrare il processo *antifrode*, mostrando il funzionamento del controllo automatizzato, i controlli di routine ed infine il processo di valutazione e azione.

2.5.1 Controllo automatizzato

Il controllo automatizzato è basato sull'implementazione R della tecnica Holt-Winters precedentemente descritta (2.3.4), eseguito su un elevato numero di serie storiche, con stagionalità supposta settimanale e generate partizionando i dati relativi alle transazioni di tipo payment-out effettuati durante gli ultimi N giorni (con N fissato arbitrariamente a 30) nel sistema; il processo può essere riassunto nei seguenti punti:

- individuazione di tutte le possibili coppie (nazione, commerciante) per le quali è avvenuto almeno un pagamento negli ultimi N giorni;
- per ognuna delle coppie individuate:
 - estrarre i dati di payment-out (in particolare il GTV in €) degli ultimi N giorni;
 - predire l'N-esimo valore mediante un modello Holt-Winters il cui training viene effettuato utilizzando gli $N - 1$ valori precedenti, l'elaborazione restituirà un intervallo con livello di confidenza al 95%;
 - verificare se il valore effettivo è incluso nell'intervallo, qualora il valore *esca* dall'intervallo di confidenza per un valore superiore a K (con K fissato arbitrariamente a 50 €) sollevare un allarme mediante SMS ed e-mail.
- generazione di un rapporto dettagliato con tutte le previsioni eseguite e invio tramite e-mail.

2.5.2 Controlli di routine

Oltre all'analisi automatizzata delle serie storiche dei payment-out vengono effettuati dei controlli di routine: l'analisi degli utenti che cancellano il proprio account, quella degli utenti che rimuovono numero di telefono (MSI-SDN) precedentemente registrati dal proprio account e il valore totale delle transazioni transfer-in rispetto a quelle di tipo payment-out.

Deleted Users

L'operazione di cancellazione di un account è un evento raro per quanto riguarda gli utenti legittimi, mentre è frequente tra i truffatori, perché è un meccanismo per aggirare azioni di blocco, per questo è opportuno analizzare il comportamento di questi utenti;

Deleted MSISDNs

Allo stesso modo la rimozione di un MSISDN dall'account di un utente è una operazione sospetta, viene quindi effettuato un controllo sul comportamento dei numeri telefonici rimossi e dei relativi utenti.

Transfer-in vs. Payment-out

Per ogni nazione viene esaminata su base giornaliera la differenza algebrica e relativa tra l'importo totale delle transazioni di tipo transfer-in e il totale di quelle di tipo payment-out; una piccola discrepanza (soprattutto nelle nazioni in cui è possibile accumulare credito senza spenderlo immediatamente) è fisiologica, ma quando questa è considerevole può essere indice di una attività fraudolenta in corso.

2.5.3 Analisi e azione

Alla luce degli alert e dei controlli di routine l'esperto di dominio:

- verifica l'eventuale attivazione di campagne promozionali da parte del merchant;
- esamina, tramite il pannello di controllo, la lista di transazioni per la coppia (nazione, commerciante)
- cerca manualmente la presenza di comportamenti anomali, ad esempio:
 - un elevato numero di transazioni per singolo numero telefonico;
 - un elevato numero di transazioni per indirizzo IP.

- decisione di blocco: quando il sospetto di frode è alto l'esperto decide di attuare delle azioni di blocco, queste possono essere relative a:
 - blocco dell'account di uno o più buyer, inibendo la possibilità di pagare con tutti i credit device (numero di telefono o linea internet nel caso di ISP billing) ivi registrati; dato che la registrazione è un'operazione facoltativa, nel caso di buyer non registrati, l'azione di blocco sarà relativa ai singoli credit device sospetti;
 - blocco di un merchant in una nazione;
 - blocco totale di un merchant;
 - blocco totale di un credit provider;
 - blocco di un'intera nazione.

A parte il primo caso, questo genere di blocco è normalmente di carattere temporaneo, nell'ottica di investigazione e risoluzione del problema che ha permesso l'attività fraudolenta.

Qualora invece il valore effettivo sia al di sotto della previsione vengono effettuate indagini volte ad individuare eventuali guasti nella filiera che va dalla pagina di pagamento fino ai sistemi del credit provider.

2.6 Esempi reali

Andremo ora ad illustrare alcuni specifici episodi avvenuti nella storia del PSP esaminando l'andamento della serie storica in un intorno temporale di alcune settimane evidenziando il valore effettivo in relazione con il valore predetto.

2.6.1 Struttura dei grafici

Per una migliore comprensione degli esempi è necessario illustrare la struttura e le caratteristiche delle tre tipologie dei grafici che saranno tracciati per

dettagliare le caratteristiche delle serie storiche e il funzionamento dell'algoritmo. Ricordiamo che le serie storiche esaminate hanno granularità giornaliera, per questo sulle ascisse di tutte e tre le tipologie di grafico abbiamo la data di riferimento. Ogni punto del grafico rappresenta una esecuzione indipendente dell'algoritmo avviata "portando indietro le lancette dell'orologio" alla data di riferimento.

Serie storica e predizione

Il primo dei tre tipi di grafico illustra l'andamento della serie storica, la linea rossa rappresenta il valore effettivo del GTV espresso in € a prescindere dalla valuta in cui viene effettuata la transazione, mentre le barre verdi rappresentano l'intervallo predetto dall'algoritmo Holt-Winters, con livello di confidenza al 95%; la X rappresenta il centro dell'intervallo. Questa tipologia di grafico sarà prodotta per tutti gli esempi.

Errore di predizione

La seconda tipologia di grafico rappresenta in rosso la stessa linea del primo, mentre in verde viene rappresentato lo scostamento **con segno** tra la linea rossa e il centro dell'intervallo di predizione (la X) del precedente grafico. Questa tipologia di grafico sarà prodotta per un sottoinsieme degli esempi.

Parametri dell'algoritmo

Il terzo ed ultimo grafico rappresenta l'andamento dei valori α , β e γ che, ricordiamo, sono determinati automaticamente dall'ambiente R minimizzando il valore dell'errore SSE con l'algoritmo **L-BFGS-B**. Questa tipologia di grafico sarà prodotta solo per il primo esempio.

2.6.2 Veri positivi

Apriamo la sequenza di esempi con alcune frodi realmente avvenute per le quali era stato correttamente attivato l'allarme.

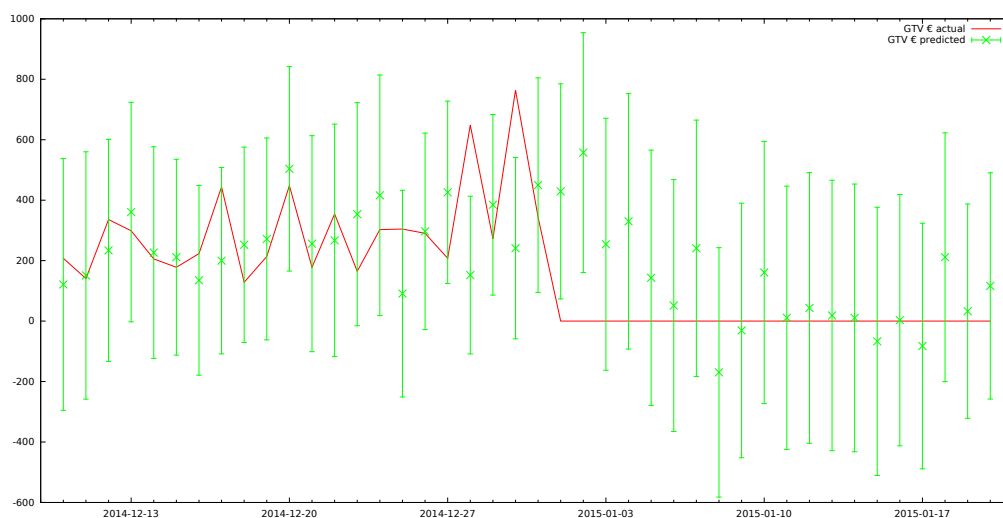


Figura 2.2: Vero positivo con blocco del merchant (caso 1)

Caso 1

Il primo caso di vero positivo che andiamo ad esaminare è relativo alla fine di dicembre 2014, quando, a causa di un guasto, gli utenti di una intera nazione riuscivano ad inviare messaggi senza che venisse loro addebitato l'importo (flusso mob-mt cfr. 1.2.4).

Il grafico in figura 2.2 illustra in rosso la serie storica giornaliera del GTV espresso in € del principale merchant della nazione, mentre le barre verdi rappresentano l'intervallo con livello di confidenza al 95% del valore predetto dall'algorithm.

Si noti la presenza di un primo picco in data 28 dicembre che fuoriesce dall'intervallo di predizione e di un secondo picco, più alto, in data 30 dicembre; i picchi sono dovuti all'attività anomala, mentre il valore basso del giorno 31 e i successivi valori a zero evidenziano l'azione di blocco presa al fine di arginare la problematica in attesa della risoluzione tecnica del problema.

Un secondo grafico, in figura 2.3, ripropone sempre in rosso la stessa serie storica della figura precedente affiancata questa volta dall'errore di predizione.

Per una panoramica completa, dato che si tratta della prima serie storica

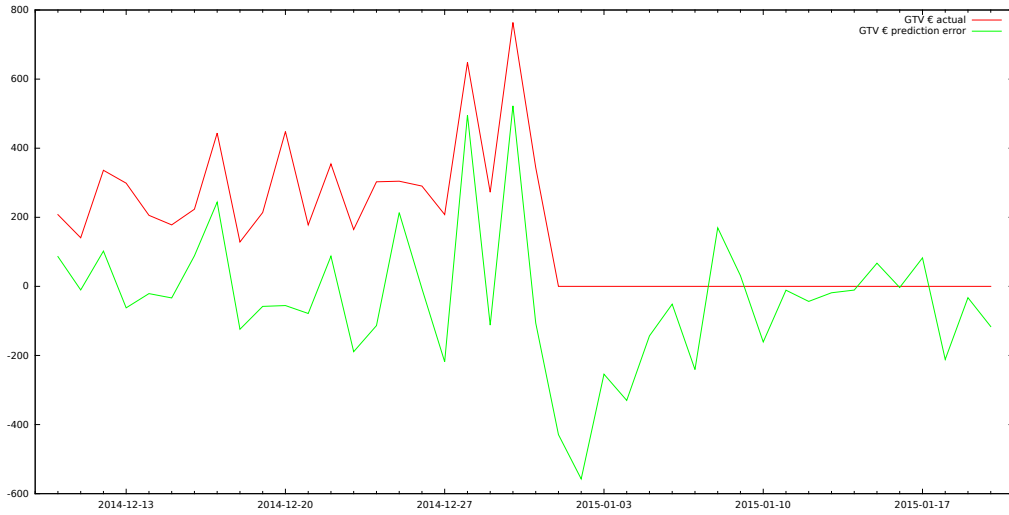
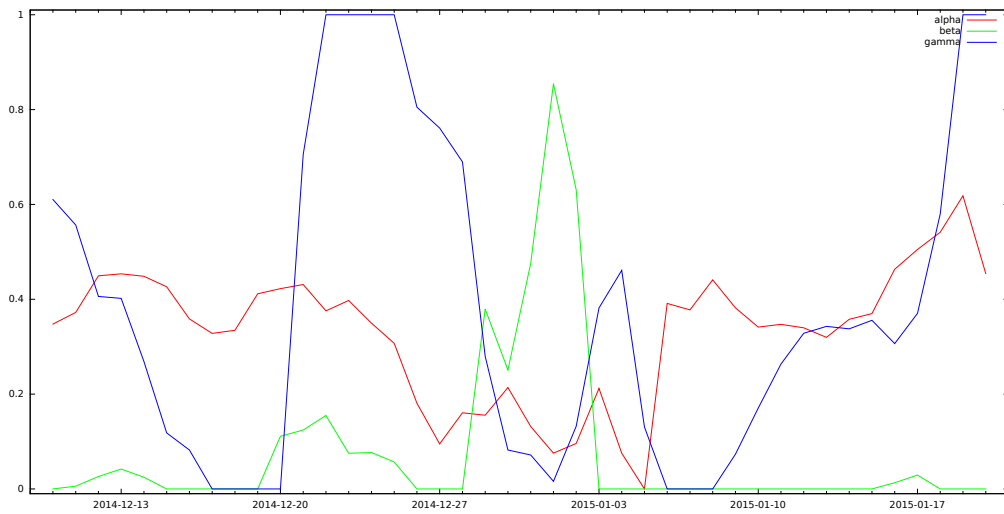


Figura 2.3: Andamento dell'errore di predizione (caso 1)

Figura 2.4: Andamento dei parametri α , β e γ (caso 1)

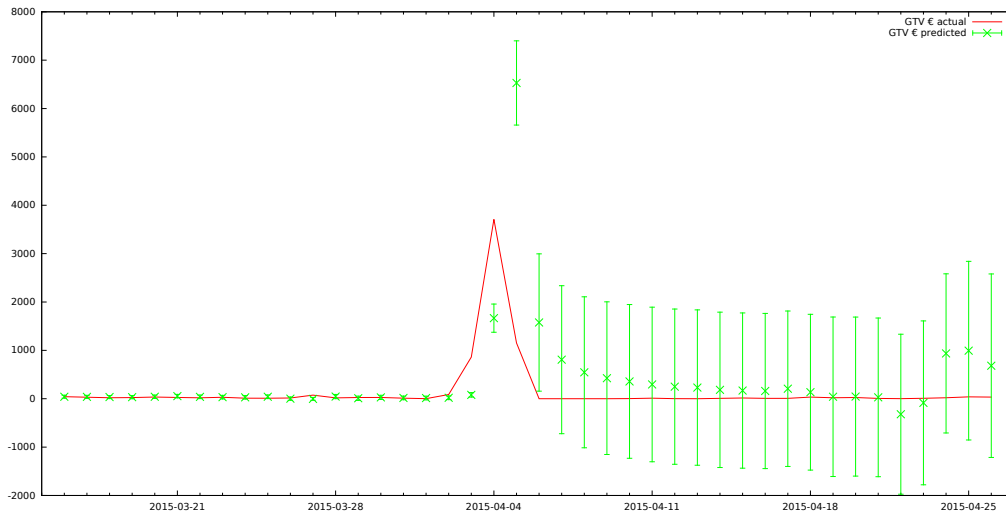


Figura 2.5: Vero positivo (caso 2)

che andiamo ad analizzare aggiungiamo un terzo ed ultimo grafico (figura 2.4) che mostra l'andamento dei parametri α , β e γ . Lo scopo di questo grafico è di evidenziare come questi parametri varino in maniera molto brusca, questo è dovuto alla durata troppo breve della serie storica data in ingresso all'algorithmo che, come già detto, è di soli 30 giorni.

Caso 2

Il secondo caso di vero positivo che andiamo ad analizzare è quello di un merchant il cui volume di transazioni in GTV, in condizioni normali, era di alcune decine di € al giorno. Nel grafico della serie temporale del GTV in figura 2.5 vediamo come in data 3 e soprattutto 4 aprile 2015 si abbia un picco di notevole ampiezza, di oltre 3700 €.

Questo picco è stato generato da un problema nei sistemi del PSP che ha permesso agli utenti di completare transazioni senza avere credito disponibile. L'intera nazione coinvolta è stata bloccata per alcuni giorni fino alla risoluzione del problema.

È interessante notare come l'anomalia influenzi notevolmente l'andamento delle previsioni future, in particolare in data 5 aprile 2015, cosa ancora più evidente analizzando l'errore di predizione (figura 2.6).

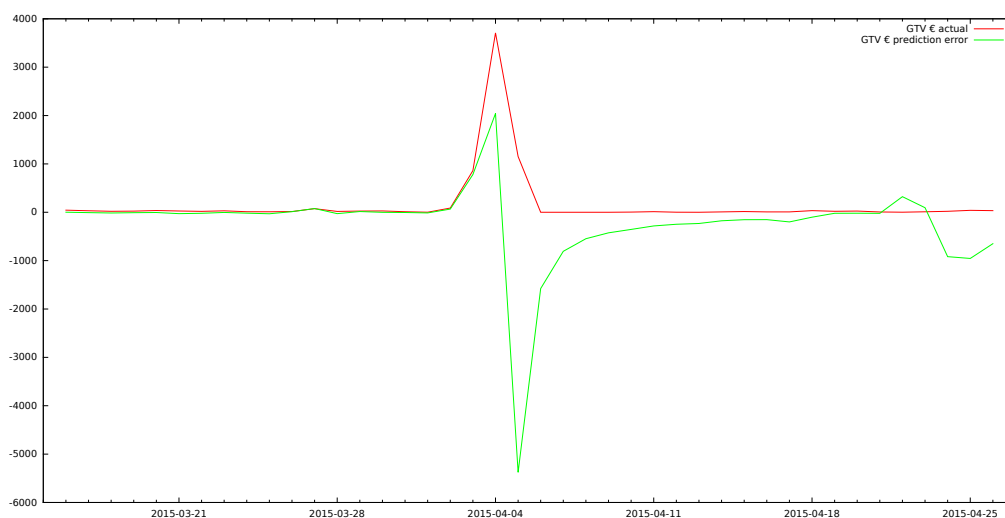


Figura 2.6: Andamento dell'errore di predizione (caso 2)

2.6.3 Falsi positivi

I casi di falso positivo sono i più comuni, abbiamo selezionato due esempi significativi per illustrarne le caratteristiche.

Caso 1

Il primo caso di falso positivo è quello nelle figure 2.7 e 2.8, si veda come in data 1 settembre 2015 si abbia una fuoriuscita dall'intervallo di predizione di circa 166 €, questo è dovuto ad una componente di stagionalità mensile non modellata e dovuta principalmente all'azzeramento dei limiti di spesa (vedi sezione 1.3.3).

L'effetto di questo picco si riflette negativamente sulle previsioni successive, in particolare soprattutto in data 8 settembre (esattamente una settimana dopo il picco anomalo) ma anche, seppur in misura minore, in data 5 settembre.

Si noti inoltre che sono presenti dei valori fuori dall'intervallo in data 26 e 29 agosto e 6 settembre che non hanno scatenato l'allarme perché al di sotto della soglia di 50 €.

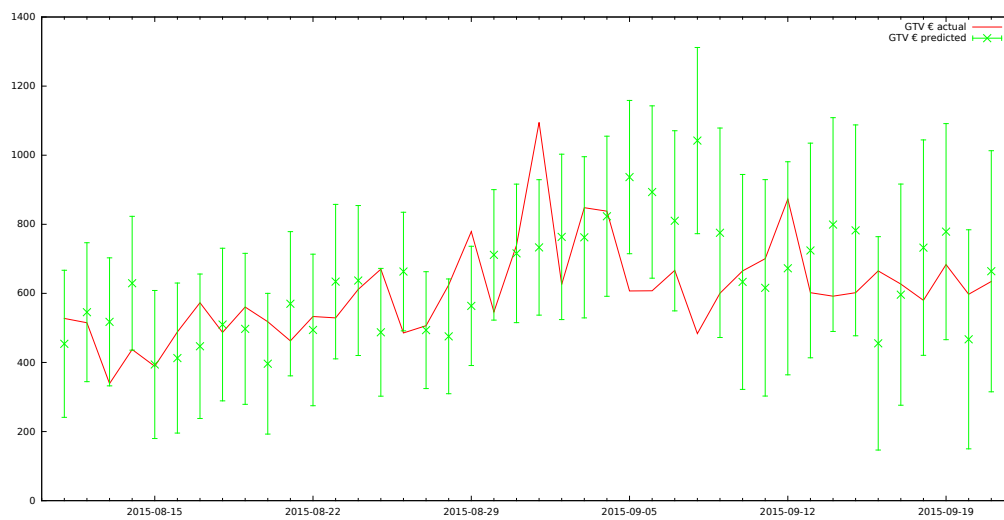


Figura 2.7: Falso positivo del 01/09 dovuto al picco di inizio mese (caso 1)

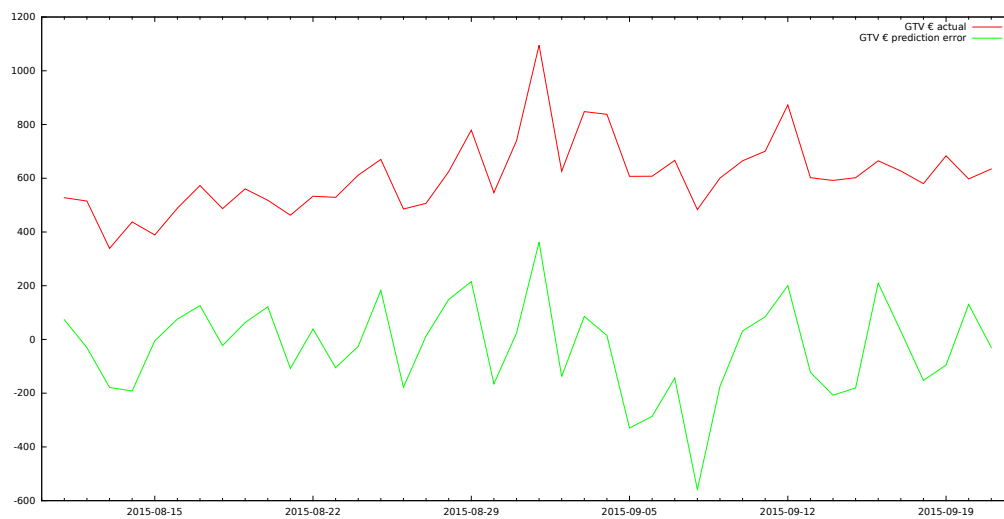


Figura 2.8: Andamento dell'errore di predizione (falso positivo caso 1)



Figura 2.9: falso positivo del 01/09 dovuto ad un picco di attivazioni (caso 2)

Caso 2

Questo secondo caso di falso positivo è invece dovuto alla componente relativa alle attivazioni dei servizi in abbonamento (subscription); le componenti (in gergo *business model*) sono divise in: pagamenti singoli (pull), attivazioni di servizio in abbonamento (subscription activation) e rinnovi del servizio (subscription renewal), le tre componenti (additive) della serie storica hanno comportamenti differenti e la loro somma è male modellata dall'algoritmo utilizzato.

Dalle figure 2.9 e 2.10 emerge la presenza di un picco in data 1 settembre 2015, si noti inoltre come in data 8 e 9 settembre si ha rispettivamente un picco negativo e positivo della serie storica, questo è dovuto ad un blocco temporaneo dei rinnovi che sono poi stati recuperati.

2.6.4 Falsi negativi

La disponibilità di esempi di falsi negativi è piuttosto scarsa, andremo ora ad analizzare alcuni episodi di mancato allarme che sarebbero dovuti scattare

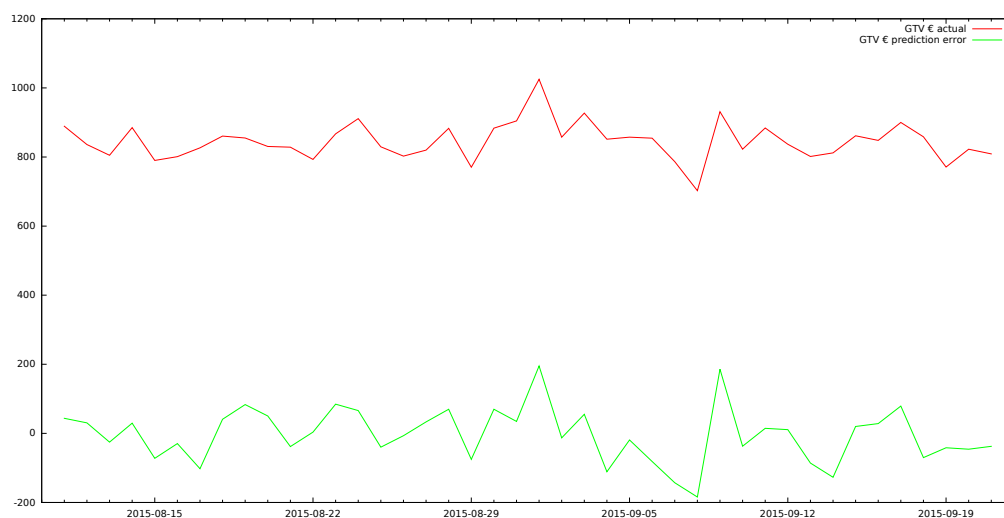


Figura 2.10: Andamento dell'errore di predizione (falso positivo caso 2)

non per una situazione fraudolenta ma per un calo del traffico durante il mese di dicembre 2014.

Caso 1

Il primo caso di falso negativo, come vediamo nella figura 2.11 è presente un calo del GTV a partire dal giorno 8 dicembre 2014 dovuto ad una riduzione del traffico, si sarebbe dovuto scatenare un alert per valore troppo basso ma questo non è accaduto.

Caso 2

Nel secondo caso di falso negativo, in maniera analoga al primo abbiamo una diminuzione del GTV a partire dal 15 dicembre 2014 come si evince dalla figura 2.12 senza che ciò causasse un alert per valore troppo basso.

2.7 Criticità

Nell'analizzare il sistema sono emerse alcune criticità, le andremo a riassumere in questa sezione:

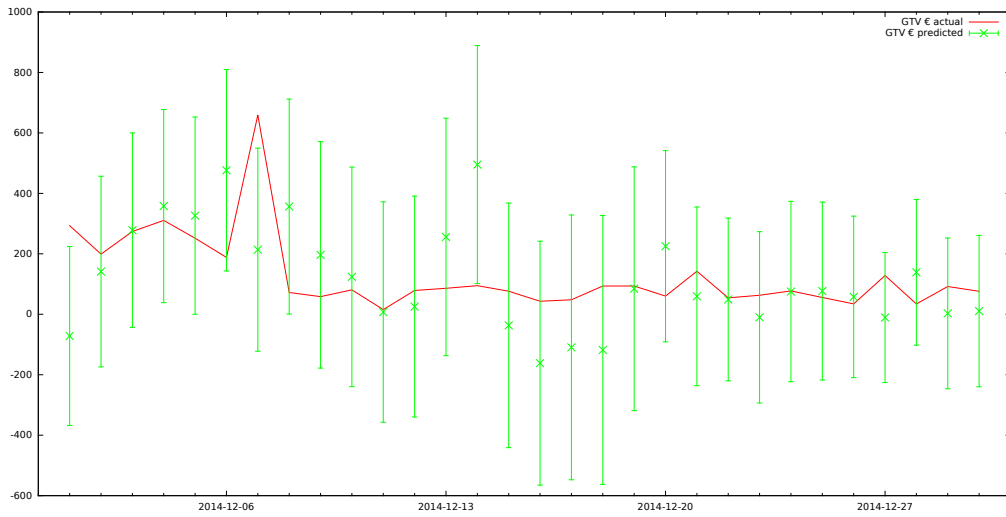


Figura 2.11: Falso negativo, calo dall'8/12 (caso 1)

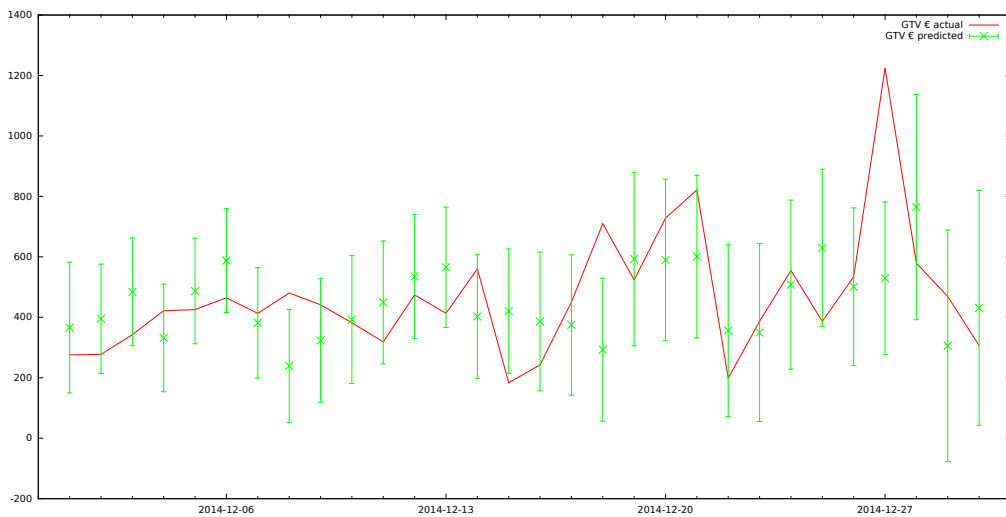


Figura 2.12: Falso negativo, calo dal 15/12 (caso 2)

- **tempo di reazione**: il controllo viene eseguito con cadenza giornaliera, una eventuale frode sarà rilevata solo al mattino successivo;
- elevato numero di **falsi positivi**: spesso gli esperti si trovano a dover fare dei controlli manuali per scoprire che non vi sono reali problemi, perdendo di fatto parecchio tempo;
- scarsa **robustezza agli outlier** [GFC10], si veda per esempio la previsione per il 5/04/2015 in figura 2.9; non è possibile inoltre, una volta accertata la frode, escludere i punti incriminati dall'algoritmo di predizione;
- la **mancata distinzione per business model**, gli acquisti singoli, le attivazioni dei servizi in abbonamento hanno dei comportamenti molto diversi e questo porta a maggiori oscillazioni dei valori;
- tutti i merchant sono trattati **allo stesso modo** indipendentemente dal loro volume di fatturato;
- difficoltà nel **distinguere una frode da una promozione**: spesso i picchi non sono dovuti ad un comportamento anomalo ma semplicemente ad una promozione;
- mancata modellazione della **stagionalità multipla**: le serie storiche non hanno solo una componente di stagionalità settimanale, ma anche una mensile ed annuale; la stagionalità annuale non sempre segue il calendario gregoriano, ad esempio in alcune nazioni come la Turchia i merchant sono soliti effettuare importanti promozioni nel periodo del Ramadan seguendo quindi il calendario islamico;
- una forte instabilità dei parametri α , β e γ , come emerge dalle figure si hanno delle forti oscillazioni dei tre parametri, probabilmente per via della dimensione troppo piccola della serie storica (30 giorni).

2.8 Conclusioni

In questo capitolo abbiamo descritto l'architettura del sistema di Onebip, approfondito la teoria delle serie storiche e in particolare lo smorzamento esponenziale triplo di Holt-Winters; siamo poi passati ad analizzare nel dettaglio il processo di fraud detection, con l'aiuto di esempi reali ne abbiamo individuato le criticità e tratto numerosi spunti per poter migliorare il sistema esistente, cosa che andremo a fare nel corso del prossimo capitolo.

Capitolo 3

Possibili miglioramenti del sistema esistente

In questo capitolo andremo ad esaminare i miglioramenti che possono essere apportati sia ai dati in ingresso sia all'algoritmo per cercare di ridurre gli errori di predizione.

Inizialmente aumenteremo il numero di giorni di training del modello Holt-Winters e successivamente valuteremo quali dimensioni di aggregazione sarebbe opportuno aggiungere.

Verrà poi discussa la possibilità di contrassegnare alcuni elementi della serie storica come anomali in modo da potere applicare uno smorzamento più aggressivo.

Il capitolo prosegue con l'implementazione e il commento di una versione robusta della tecnica Holt-Winters [GFC10].

Infine verranno trattati dei miglioramenti inerenti l'esperienza utente come ad esempio la gestione di livelli multipli di allerta.

3.1 Metriche di benchmark

Al fine di valutare le prestazioni degli algoritmi di previsione è necessario definire una metrica di benchmark. Abbiamo scelto di utilizzare la metrica

mean squared prediction error (MSPE) detta anche mean squared forecast error (MSFE) (d'ora in avanti MSPE).

In statistica l'errore quadratico medio di previsione è il valore atteso del quadrato della differenza tra gli elementi calcolati a partire dal modello di smussamento (cfr. 2.3.3) e il valore effettivo, questo è espresso dall'equazione:

$$MSPE = \frac{1}{n} \sum_{t=1}^n (\hat{y}_{t|t-1} - y_t)^2 \quad (3.1)$$

Dove:

- n è il numero di predizioni effettuate;
- y_t è il valore effettivo della serie storica all'istante t ;
- $\hat{y}_{t|t-1}$ è il valore di y_t predetto dal modello generato utilizzando gli elementi fino a $t - 1$.

3.2 Lunghezza del periodo di training

Come già detto nel capitolo 2, il periodo di training dell'algoritmo Holt-Winters utilizzato nella soluzione preesistente consiste negli ultimi 30 giorni della serie storica del GTV in € di un determinato merchant ristretto ad una specifica nazione.

La periodicità della serie storica del caso d'uso del PSP è di 7 giorni e che quindi 30 elementi rappresentano poco più di 4 *stagioni*.

Per semplicità chiameremo periodo di training l'intero intervallo di date precedente alla previsione anche se è opportuno ricordare che le prime due stagioni vengono utilizzate per determinare i valori iniziali e che quindi l'algoritmo vero e proprio viene eseguito scartando la prima stagione.

Prenderemo quindi alcuni degli esempi mostrati nel capitolo 2 e valuteremo la differenza della metrica di benchmark all'aumentare della lunghezza del periodo di training e valutando il giusto compromesso tra precisione e complessità spaziale e temporale.

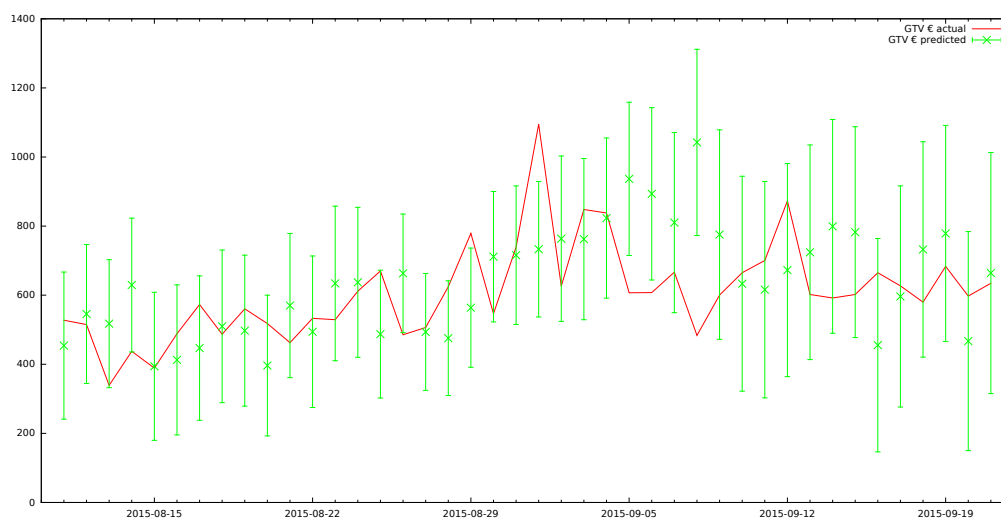


Figura 3.1: Falso positivo caso 1, predizione con training su 30 elementi

3.2.1 Falso positivo, caso 1

Riproponiamo il falso positivo n° 1 già visto in sezione 2.6.3, dove in data 1 settembre 2015 si è avuto un picco dovuto alla stagionalità mensile; vediamo come rispetto alla figura 3.1, l'allungamento del periodo di training da 30 a 100 elementi (figura 3.2) non risolve il falso positivo, ma riduce notevolmente l'effetto dell'outlier di data 1 settembre sulla settimana successiva (8 settembre).

Il falso positivo permane a causa della mancata modellazione della stagionalità multipla (viene modellata solamente quella settimanale ma non quella mensile).

Un ulteriore allargamento del periodo di training a 200 elementi (figura 3.3) comporta un beneficio di piccola entità non facilmente apprezzabile dal grafico.

Parametri di smorzamento

L'andamento dei parametri di smorzamento che, ricordiamo, sono determinati dall'algoritmo al fine di minimizzare l'errore quadratico totale di predizione (ogni punto del grafico corrisponde ad una differente esecuzione

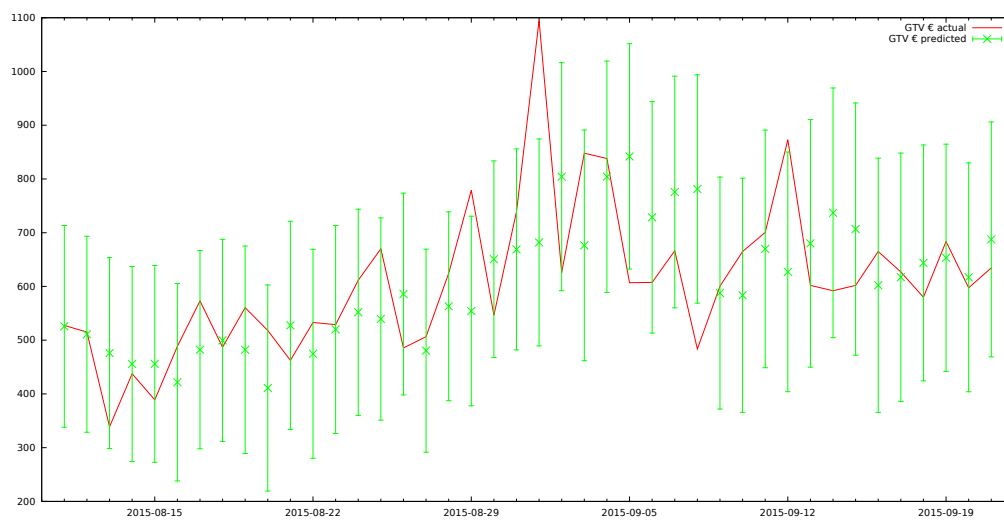


Figura 3.2: Falso positivo caso 1, predizione con training su 100 elementi

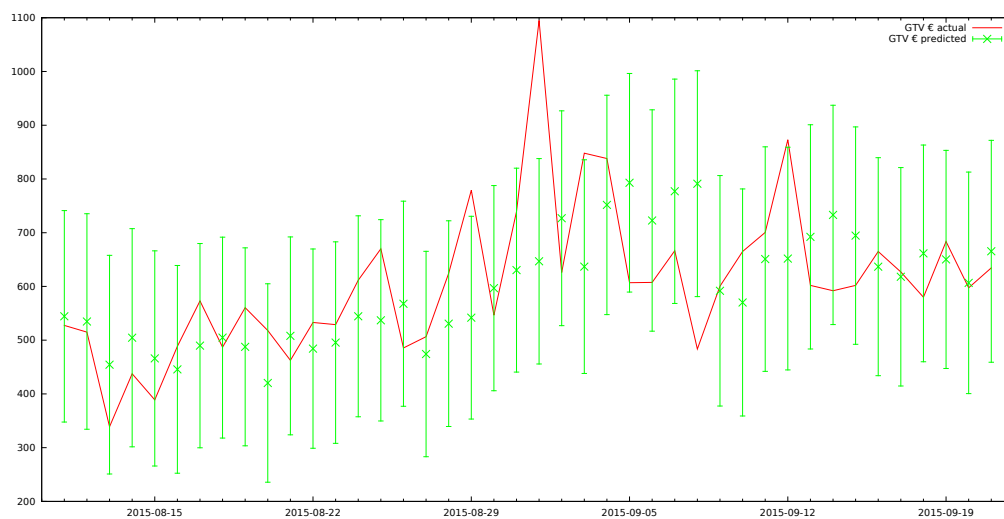


Figura 3.3: falso positivo caso 1, predizione con training su 200 elementi

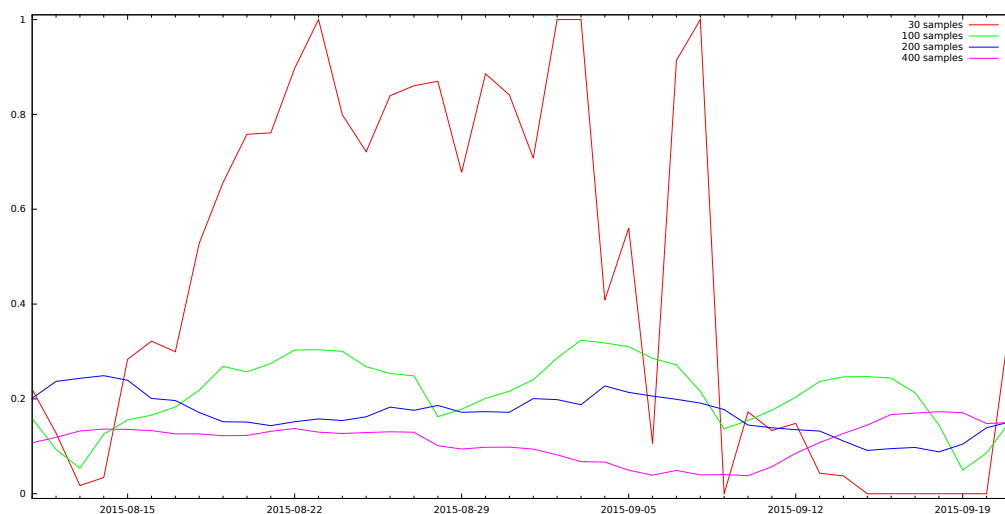


Figura 3.4: andamento del parametro α al variare della lunghezza del periodo di training

ne dell'algoritmo) fornisce un'indicazione qualitativa di quanto rapidamente varino le caratteristiche della serie storica in esame, tenendo presente che variazioni troppo brusche sono spesso indice di **overfitting**.

In figura 3.4 abbiamo l'andamento del parametro α per quattro diverse lunghezze del periodo di training: 30, 100, 200 e 400.

In rosso abbiamo l'andamento del parametro α con un periodo di training lungo 30 giorni, dal grafico si nota come questo sia piuttosto instabile nel tempo e come raggiunga valori vicini o uguali agli estremi. Questo causa uno smorzamento rapidissimo della componente di livello del modello quando il valore si avvicina ad 1.0 ovvero la previsione sarà influenzata solo dai valori recenti delle serie storica. Quando il valore si avvicina a 0.0 la componente di livello, non viene più influenzata dai valori recenti ma rimane pressoché costante al valore iniziale.

In verde abbiamo l'andamento del parametro α con un periodo di training lungo 100 giorni, vediamo come il valore si mantenga tra circa 0.1 e 0.4 senza cambiamenti troppo bruschi e senza raggiungere i valori estremi. Questo significa che la componente della serie di livello sarà influenzata in maniera equilibrata dagli elementi recenti e quelli meno recenti, secondo una

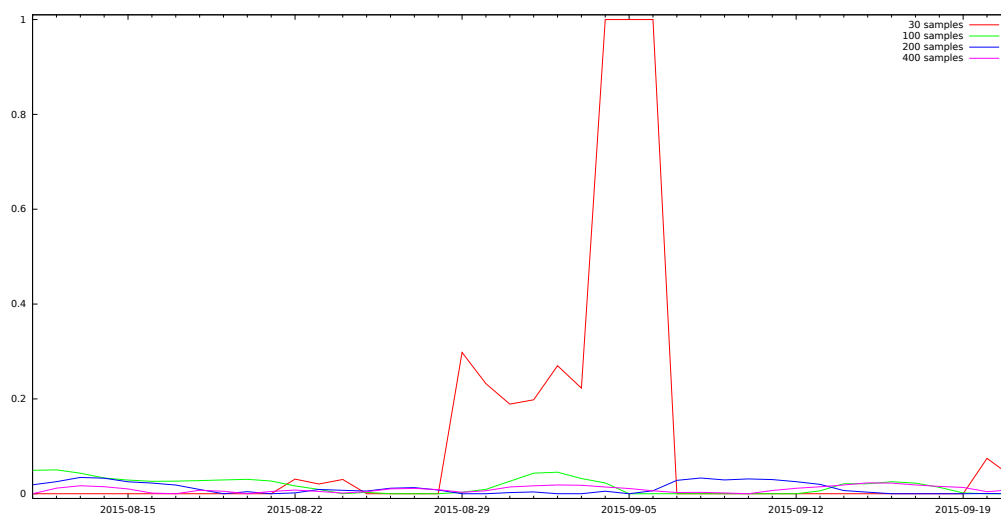


Figura 3.5: andamento del parametro β al variare della lunghezza del periodo di training

legge esponenziale.

Nelle linee in blu e in magenta il periodo di training è stato aumentato ulteriormente e portato rispettivamente a 200 e 400 elementi, il miglioramento dell'andamento non giustifica il maggior sforzo computazionale necessario per effettuare le predizioni.

In figura 3.5 abbiamo l'andamento del parametro β per quattro diverse lunghezze del periodo di training: 30, 100, 200 e 400.

In rosso abbiamo l'andamento del parametro β con un periodo di training di 30 giorni, dal grafico si nota come questo si mantenga tra 0.0 e 0.1 fino al 28/08/2015 dove inizia a variare bruscamente per poi saltare ad 1.0 rinunciando quindi a smussare la componente di trend e poi nuovamente dal 07/09/2015 bruscamente a 0.0 mantenendo quindi costante la componente di trend.

In verde abbiamo l'andamento del parametro β con un periodo di training lungo 100 giorni, vediamo come il valore si mantenga tra circa 0.0 e 0.1 senza cambiamenti bruschi, indice del fatto che la componente di trend cambia lentamente.

Come per α nelle linee in blu e in magenta il periodo di training è stato

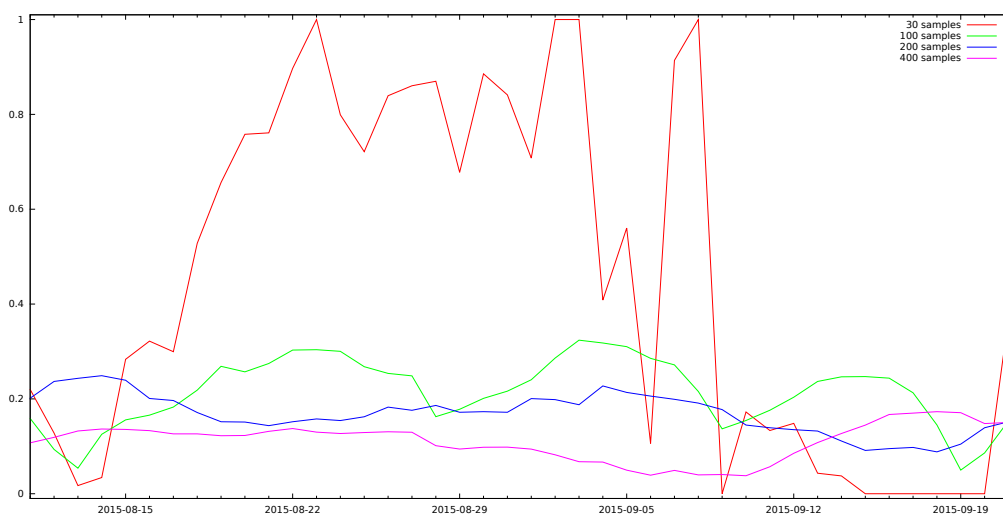


Figura 3.6: andamento del parametro γ al variare della lunghezza del periodo di training

aumentato ulteriormente e portato rispettivamente a 200 e 400 elementi, il miglioramento dell'andamento anche questa volta non giustifica il maggior sforzo computazionale necessario per effettuare le predizioni.

In figura 3.6 abbiamo l'andamento del parametro γ per quattro diverse lunghezze del periodo di training: 30, 100, 200 e 400.

In rosso abbiamo l'andamento del parametro γ con un periodo di training di 30 giorni, dal grafico si nota come questo partendo da valori prossimi allo zero salga rapidamente e si stabilizzi su valori alti, segno del fatto che l'algoritmo non individui alcuna stagionalità e per questo smussi molto poco questa componente. L'algoritmo fatica ad individuare la componente di stagionalità già a partire dal 22/08/2015, in data quindi antecedente all'outlier; questo è dovuto al fatto che la stagionalità del PSP è settimanale e che con un periodo di training di 30 giorni si hanno a disposizione poco più di 4 stagioni senza contare che la prima stagione viene esclusa dalla computazione perché necessaria al calcolo dei coefficienti iniziali.

In verde abbiamo l'andamento del parametro γ con un periodo di training lungo 100 giorni, vediamo come il valore si mantenga tra circa 0.2 e 0.4 senza cambiamenti bruschi, indice del fatto che la componente di stagionalità cambi

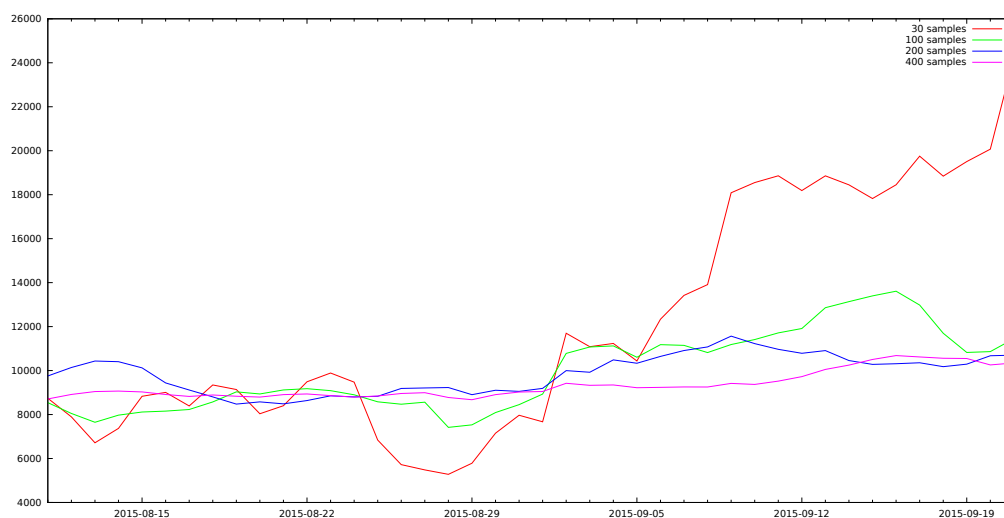


Figura 3.7: Andamento del valore di MSPE

lentamente.

Come per α e β nelle linee in blu e in magenta il periodo di training è stato aumentato ulteriormente e portato rispettivamente a 200 e 400 elementi, il miglioramento dell'andamento anche questa volta non giustifica il maggior sforzo computazionale necessario per effettuare le predizioni.

MSPE della fase di training

Al termine della fase di training dell'algoritmo, l'implementazione R fornisce il valore della somma quadratica degli errori di predizione SSE, dalla quale, dividendo per la lunghezza del periodo di training (escludendo la prima stagione), si ricava l'errore medio quadratico di predizione (MSPE), la figura 3.7 rappresenta l'andamento dell'errore MSPE che ci da un'indicazione qualitativa sulla bontà del modello, mentre per la valutazione quantitativa rimandiamo alla sezione 3.2.2.

In rosso abbiamo l'andamento dell'errore con un periodo di training lungo 30 giorni, dal grafico vediamo come l'errore si mantenga a livelli accettabili nella prima parte del grafico per poi arrivare a livelli piuttosto alti nella seconda parte, indice del fatto che il modello peggiori dal punto di vista qualitativo.

In verde abbiamo l'andamento dell'errore con un periodo di training lungo 100 giorni, si vede come questo sia più stabile. In alcuni casi il modello allenato con 100 giorni sembra mostrare un MSPE maggiore rispetto al modello allenato con 30 giorni, ma sull'intero periodo di analisi il modello allenato con 100 giorni non degenera mai, riuscendo quindi a fornire previsioni in media migliori.

Nelle linee in blu e in magenta il periodo di training è stato aumentato ulteriormente e portato rispettivamente a 200 e 400 elementi, il miglioramento dell'andamento non giustifica il maggior sforzo computazionale necessario per effettuare le predizioni.

3.2.2 MSPE della fase di predizione

Al fine di ottenere una misura quantitativa delle performance dell'algoritmo di predizione si è deciso, per ognuno degli esempi illustrati nel capitolo 2 di selezionare un intervallo di 100 date, centrato sulla data dell'evento in esame (vero positivo, falso positivo o falso negativo) e “portando indietro le lancette dell'orologio” ad ognuna di queste date calcolata la somma dei quadrati degli errori di predizione (MSPE).

Sia t_e l'istante in cui è avvenuto l'evento definiamo in questo contesto la misura MSPE come segue:

$$MSPE = \frac{1}{n} \sum_{t=t_e-50}^{t_e+49} (\hat{y}_{t|t-1} - y_t)^2 \quad (3.2)$$

Occorre specificare che ogni previsione \hat{y}_t viene ottenuta da un'esecuzione indipendente dell'algoritmo Holt-Winters (e di conseguenza con un diverso training set).

In tabella 3.1 notiamo come, aumentando il periodo di training da 30 a 100 giorni, in tutti e 6 i casi si ha un netto miglioramento dell'errore, mentre aumentando ulteriormente il numero di giorni, le prestazioni migliorano di poco o, in alcuni casi, peggiorano.

Caso	MSPE			
	30 gg.	100 gg.	200 gg.	400 gg.
Vero positivo 1	33507	22214	21626	21914
Vero positivo 2	405048	388379	390246	453585
Falso positivo 1	27052	15951	16019	15594
Falso positivo 2	5515	3718	3554	3325
Falso negativo 1	20546	15732	14608	14566
Falso negativo 2	35937	24120	21335	21025

Tabella 3.1: Riepilogo delle metriche di errore sui casi illustrati

Per questo motivo abbiamo dimostrato che il giusto compromesso tra la lunghezza del periodo di training (che incide sulla complessità computazionale) e le prestazioni dell'algoritmo è di 100 giorni.

3.3 Ulteriori dimensioni di aggregazione

La recente evoluzione del sistema di reportistica (2.1) ha permesso la raccolta di dati molto più dettagliati rispetto a quando il processo antifrode era stato messo a punto per la prima volta, andremo quindi a valutare ulteriori suddivisioni dei dati in aggiunta a quella per nazione e merchant.

3.3.1 Business model

Per business model si intende la distinzione tra pagamenti **una tantum** (pull), primo pagamento all'atto dell'attivazione di servizi in abbonamento (subscription activation) e pagamenti relativi al rinnovo (subscription renewal). Il comportamento di queste tre componenti è molto diverso, in particolare:

- **pull**: l'utente effettua un singolo pagamento, la transazione può avere successo o fallire definitivamente;
- **subscription activation**: l'attivazione di un servizio in abbonamento utilizza logiche differenti, in caso di fallimento è possibile tentare nuo-

vamente l'addebito, che potrebbe avvenire anche a distanza di giorni da quando l'utente ha espresso la volontà di iscriversi al servizio;

- **subscription renewal**: l'andamento di queste transazioni non dipende dalle azioni dell'utente (salvo la disdetta dell'iscrizione); può accadere che i rinnovi vengano sospesi a causa di un guasto e che questi vengano recuperati il giorno successivo causando dei picchi anomali.

Per effettuare un'analisi qualitativa di questa suddivisione andiamo quindi a riesaminare il falso positivo n° 2 (cfr. 2.6.3) aggiungendo come dimensione di aggregazione il business model, ne derivano quindi tre serie storiche. Per tutte le previsioni utilizzeremo un periodo di training di 100 giorni.

In figura 3.8 abbiamo la serie storica del solo business model **pull** dell'esempio in oggetto, che è quello maggiormente soggetto a frodi. Notiamo come la serie non mostri alcun picco anomalo in data 01/09/2015 e si mantenga all'interno dell'intervallo di predizione, questo significa che, per quanto riguarda questo sottoinsieme dei dati, siamo di fronte ad un caso di vero negativo. Il poter escludere anomalie in questo business model è di particolare interesse e valore per il business proprio per via della distribuzione delle frodi.

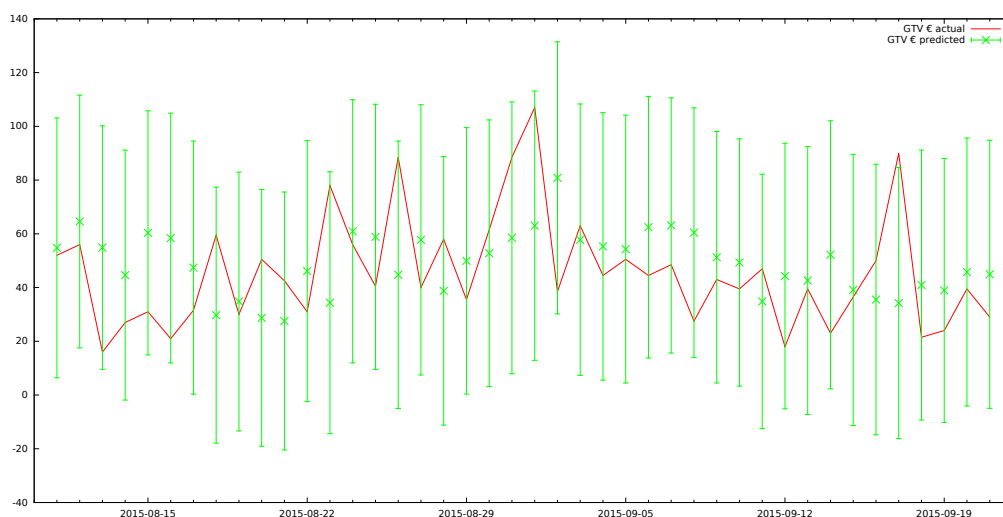


Figura 3.8: Falso positivo n° 2 cfr. sezione 2.6.3, GTV € giornaliero, business model **pull**

In figura 3.9 possiamo esaminare la componente **subscription activation** e confermare che l'origine del falso positivo è dovuta ad un picco di attivazioni in data 01/09/2015, isolando quindi la causa dell'anomalia. Questo business model è poco soggetto a frodi.

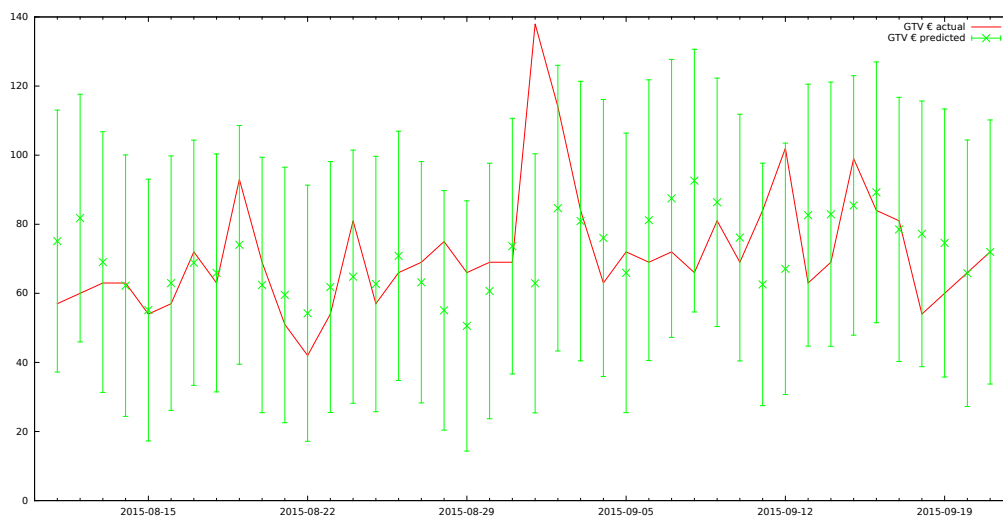


Figura 3.9: Falso positivo n° 2 cfr. sezione 2.6.3, GTV € giornaliero, business model **subscription activation**

In figura 3.10 possiamo esaminare la componente **subscription renewal** dove vediamo che il trend è costante, non vi è il picco anomalo del 01/09/2015 anche se in data 7 e 8 settembre vi è un drastico calo dei rinnovi, dovuto a un problema tecnico con successivo recupero in data 9 settembre che causa un nuovo picco. Anche questo business model è poco soggetto a frodi.

Concludiamo che questa divisione non elimina i picchi di falso positivo, ma permette di identificarne meglio l'origine, dato che quasi mai gli episodi fraudolenti sono legati ai servizi in abbonamento. Inoltre per un PSP poter escludere due business model dalla fase di investigazione consente di risparmiare parecchie risorse.

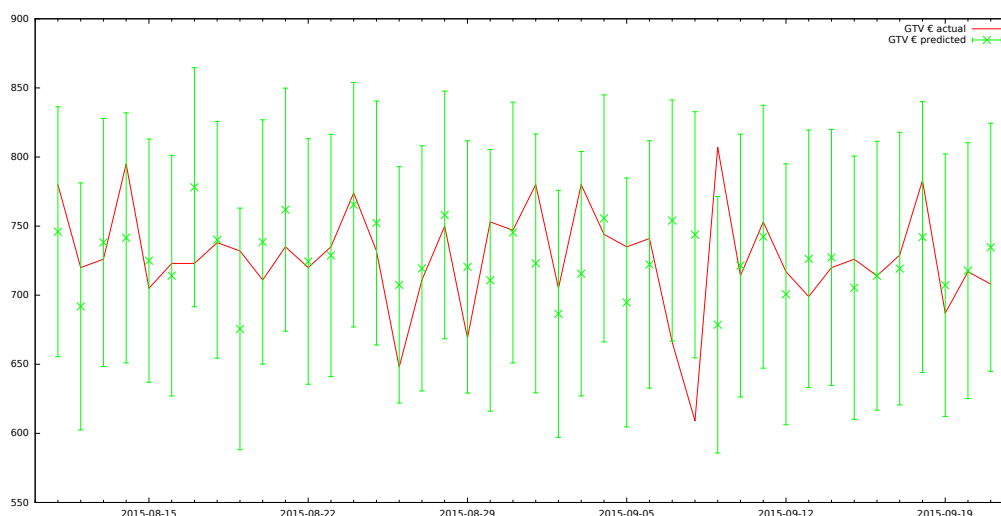


Figura 3.10: Falso positivo n° 2 cfr. sezione 2.6.3, GTV € giornaliero, business model **subscription renewal**

3.3.2 Credit provider

Spesso può capitare che le frodi riguardino utenti di un solo credit provider, avendo una ulteriore dimensione di aggregazione può essere utile ad individuare più facilmente l'origine dell'attività fraudolenta.

Riprendiamo ora il caso positivo n° 1 visto in sezione 2.6.2 dividendo la serie storica per credit provider. In figura 3.11 abbiamo il grafico della serie storica nella sua interezza, andremo ora a scomporla nei tre credit provider.

Analizzando il grafico in figura 3.12 vediamo come il primo dei credit provider non è coinvolto nell'attività di frode in quanto il valore si mantiene all'interno dell'intervallo di confidenza. La figura 3.13 evidenzia inequivocabilmente come i due picchi in data 28 e 30 dicembre 2014 siano stati causati dal credit provider B. Ad ulteriore conferma del fatto che l'anomalia coinvolge solo il credit provider B anche il credit provider C mostra in figura 3.12 un comportamento che rientra nell'intervallo di confidenza. Concludiamo che la suddivisione per credit provider può essere utile per velocizzare le operazioni di indagine al fine di individuare ed arginare più rapidamente il problema alla sola sorgente dell'anomalia facendo risparmiare tempo di investigazione e soprattutto di intervento.

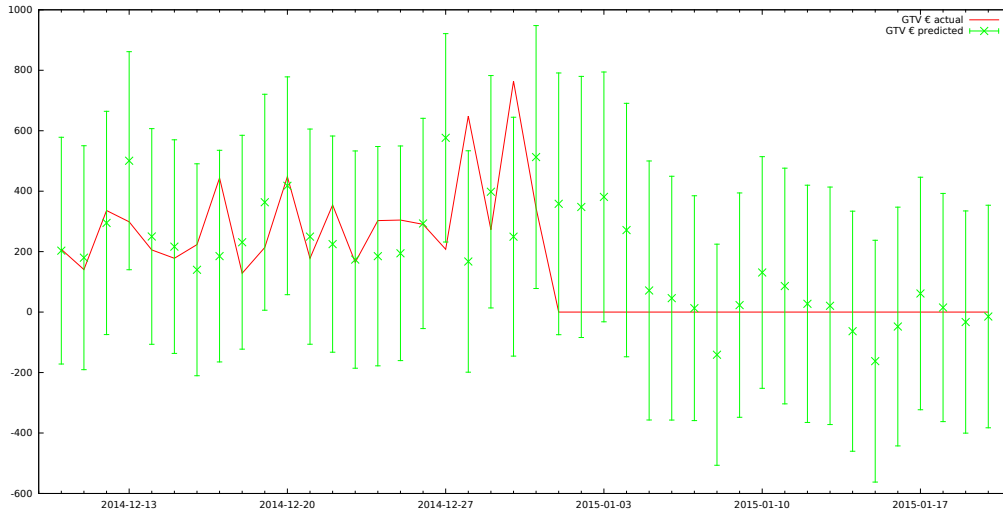


Figura 3.11: Vero positivo con blocco del merchant (caso 1) totale

In questo genere di casi è frequente che all'inizio dell'investigazione si decida per una azione proattiva di blocco del merchant all'interno della nazione coinvolta. La suddivisione per credit provider permette una azione di blocco più selettiva riducendo l'impatto sulla perdita del fatturato relativo alle transazioni legittime.

3.4 Correzione supervisionata dei dati

Un possibile approccio per il miglioramento dei falsi positivi consiste nella correzione supervisionata dei dati. Qualora l'esperto di dominio identifichi e contrassegni alcuni elementi della serie storica come anomali allora è possibile sostituire automaticamente i valori reali con dei valori smussati, in modo da non peggiorare le prestazioni delle previsioni successive.

Sia O l'insieme degli indici corrispondenti agli elementi contrassegnati come anomali da parte degli esperti di dominio. La serie prefiltrata y_t^* si calcola con la seguente formula:

$$y_t^* = \begin{cases} \hat{y}_{t|t-1} & \text{se } t \in O \\ y_t & \text{altrimenti} \end{cases} \quad (3.3)$$

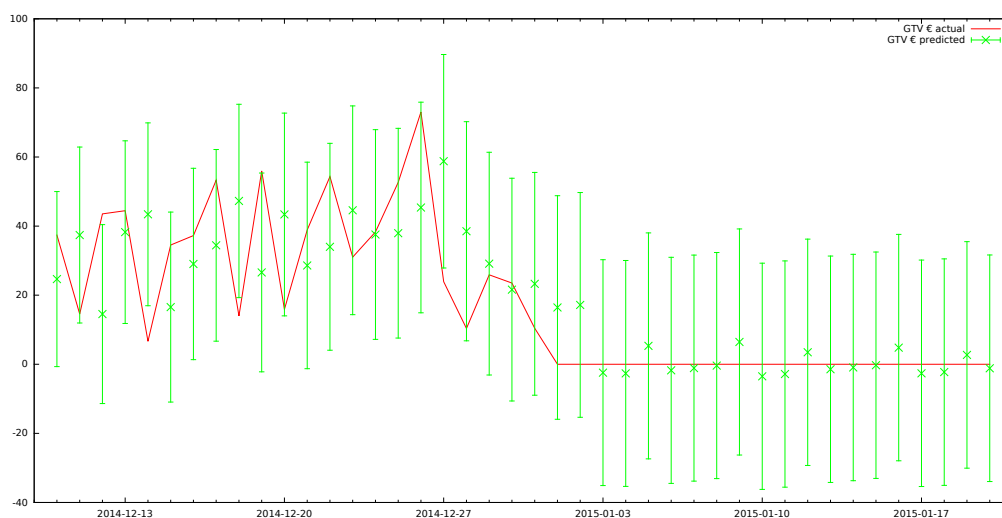


Figura 3.12: Vero positivo con blocco del merchant (caso 1) credit provider A

La quale evita di tenere in considerazione i valori precedentemente contrassegnati come anomali. Questo approccio permette di migliorare notevolmente le prestazioni di predizione sugli elementi successivi agli outlier, il vantaggio è ancora più evidente se si utilizzano periodi di training brevi, si confronti la figura 3.16 con la 3.15 facendo particolarmente attenzione alla data del 08/09/2015.

Portando il periodo di training nuovamente a 100 giorni il miglioramento è comunque apprezzabile (si confronti la figura 3.18 con la 3.17).

Per poter applicare questo accorgimento è necessario che la quantità di dati da supervisionare sia limitata. Nel nostro contesto, nonostante il sistema automatico analizzi quotidianamente oltre 600 serie storiche, il numero degli allarmi varia tipicamente da 0 a 10, rendendo possibile un approfondito controllo manuale.

Questo miglioramento è stato particolarmente apprezzato perché nonostante non abbia un fondamento teorico raffinato è decisamente pragmatico ed ha permesso al team di *reporting and control* del PSP di eliminare parecchi falsi allarmi.

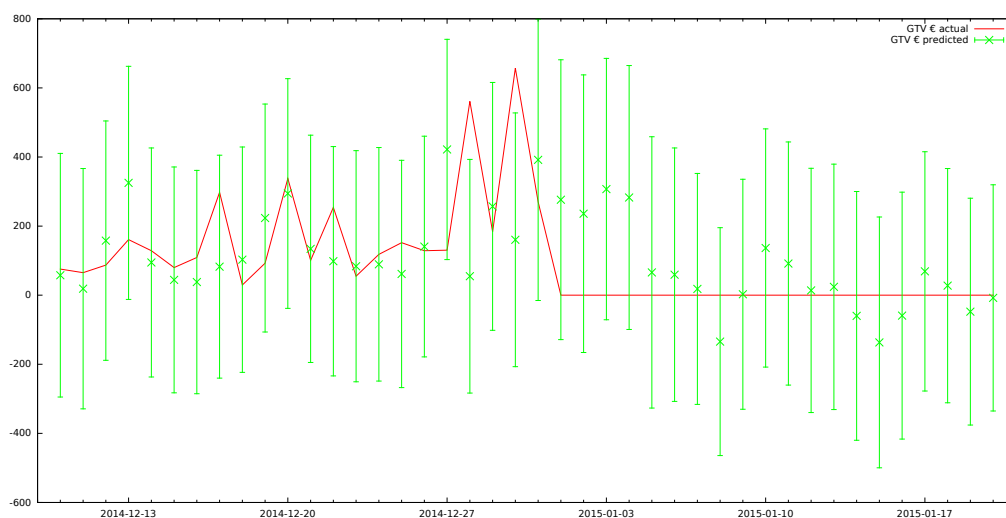


Figura 3.13: Vero positivo con blocco del merchant (caso 1) credit provider B

3.5 Miglioramento della user experience

Per una migliore esperienza utente è possibile apportare alcuni accorgimenti al fine di risparmiare risorse, sia in termini di costo dei server che in impegno del team per il controllo degli alert.

3.5.1 Granularità variabile

La suddivisione del volume di fatturato tra i vari merchant non è uniforme, ma ricorda in qualche modo il *principio di Pareto*, ovvero un numero limitato di merchant sviluppa la maggior parte del fatturato.

I merchant principali sviluppano un volume di fatturato tale da giustificare la suddivisione delle serie storiche utilizzando ulteriori dimensioni di aggregazione, mentre per quelli che sviluppano dei volumi limitati non è conveniente utilizzare lo stesso livello di dettaglio. Per questo motivo il processo può essere migliorato stabilendo delle regole basate sul volume di fatturato per decidere se sia o meno opportuno suddividere ulteriormente le serie storiche prese in esame.

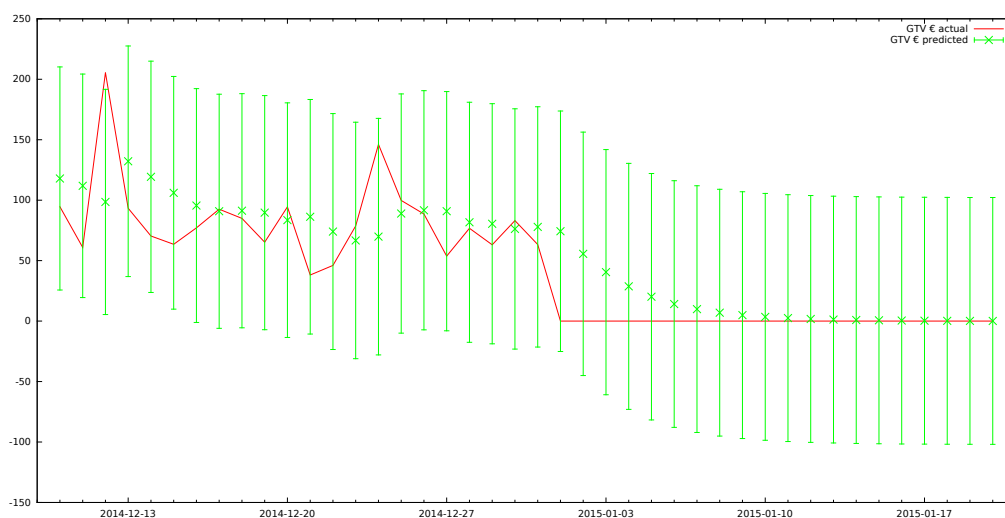


Figura 3.14: Vero positivo con blocco del merchant (caso 1) credit provider C

3.5.2 Gestione dei nuovi merchant

Può capitare che vengano siglati accordi con nuovi merchant, ed in questo caso non si hanno a disposizione dati sufficienti per generare una previsione. Per questi casi si è deciso di disattivare l'esecuzione dell'algoritmo e di procedere ad un attento monitoraggio manuale. Quanto sarà disponibile una serie storica con un numero di elementi sufficiente sarà possibile interrompere il controllo manuale e passare a quello automatico.

3.5.3 Livelli graduali di allarme

Un'altra possibilità di miglioramento pratico è quello di stabilire soglie differenti a seconda delle quali scatenare alert più o meno urgenti, utilizzando mezzi di comunicazione differenti:

- e-mail;
- chat aziendale;
- SMS;

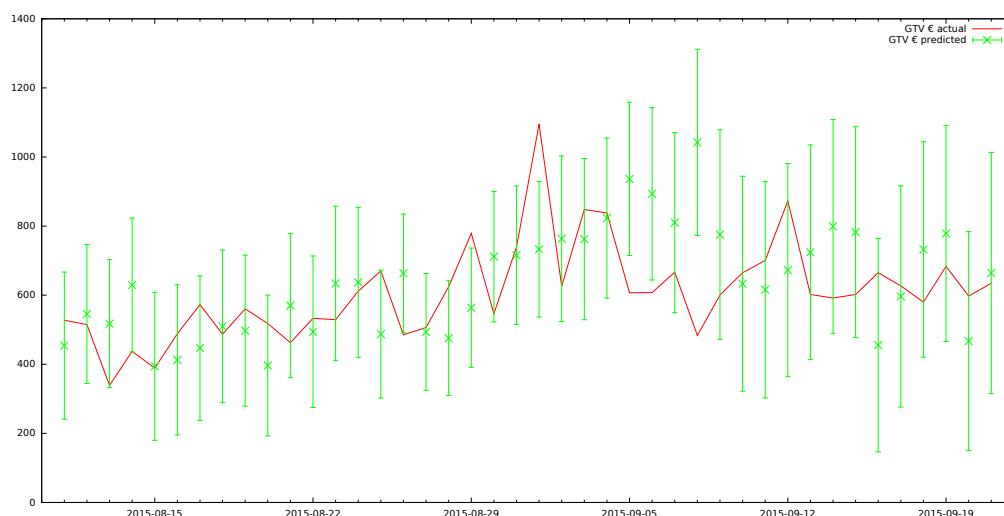


Figura 3.15: Falso positivo caso 1, predizione con training su 30 elementi

- telefonata.

Il criterio per la selezione del tipo di allarme è quello del definire soglie multiple sulla discrepanza tra il valore predetto e quello effettivo in modo da selezionare l'allarme più appropriato alla segnalazione da effettuare.

3.6 Holt-Winters Robusto

Gelper et. al [GFC10] propongono una variante robusta della tecnica Holt-Winters con stagionalità di tipo additivo (cfr. sezione 2.3.3), questa consiste nel sostituire, all'interno delle equazioni la serie y_t con una versione prefiltrata y_t^* . L'insieme delle tre equazioni della tecnica Holt-Winters diventa quindi:

$$a_t = \alpha(y_t^* - S_{t-s}) + (1 - \alpha)(a_{t-1} + F_{t-1}) \quad (3.4)$$

$$F_t = \beta(a_t - a_{t-1}) + (1 - \beta)F_{t-1} \quad (3.5)$$

$$S_t = \gamma(y_t^* - a_t) + (1 - \gamma)S_{t-s} \quad (3.6)$$

La serie prefiltrata si ottiene mediante l'equazione:

$$y_t^* = \psi \left(\frac{y_t - \hat{y}_{t|t-1}}{\hat{\sigma}_t} \right) \hat{\sigma}_t + \hat{y}_{t|t-1} \quad (3.7)$$

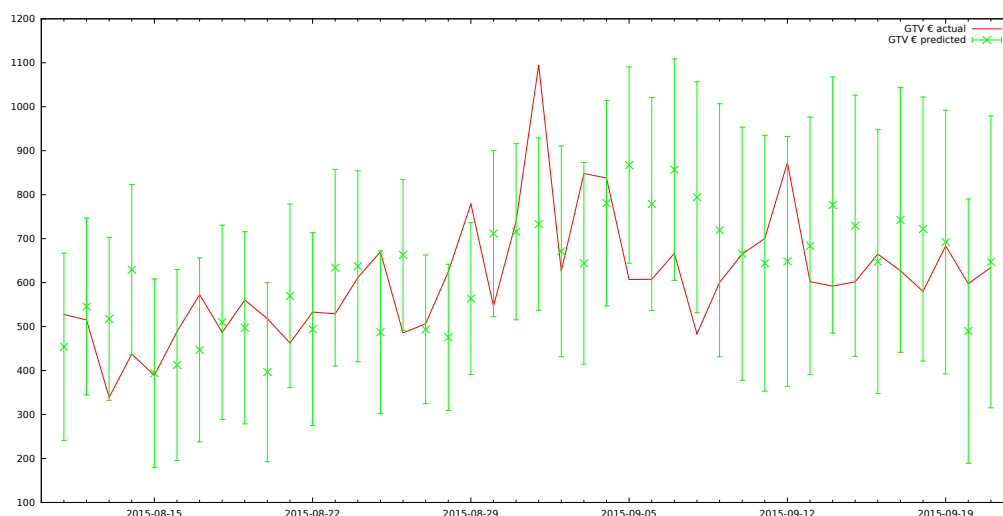


Figura 3.16: Falso positivo caso 1, predizione con Holt-Winters supervisionato e training su 30 elementi

Dove la funzione ψ di Huber è applicata all'errore di predizione $r_t = y_t - \hat{y}_{t|t-1}$ normalizzato mediante un fattore $\hat{\sigma}_t$, lo scopo di questa funzione è quella di ridurre il contributo degli outlier ed è così definita:

$$\psi(x) = \begin{cases} x & \text{se } |x| < k \\ \text{sign}(x)k & \text{altrimenti} \end{cases} \quad (3.8)$$

La fase di prefiltraggio può essere interpretata come una sostituzione dei valori inaspettatamente troppo alti o bassi con dei valori più plausibili. In particolare se l'errore di predizione r_t è piccolo il valore filtrato y_t^* coincide con y_t . D'altro canto se la deviazione tra il valore predetto e quello osservato è troppo grande l'outlier viene rimpiazzato con un valore limite che dipende dalla costante k .

La scelta comune (che verrà adottata) è $k = 2$, supponendo implicitamente che l'errore $r_t = y_t - \hat{y}_{t|t-1}$ sia distribuito normalmente. Da notare che nel caso in cui $k = +\infty$ la serie filtrata coincide con la serie originale riconducendo la tecnica alla versione classica di Holt-Winters.

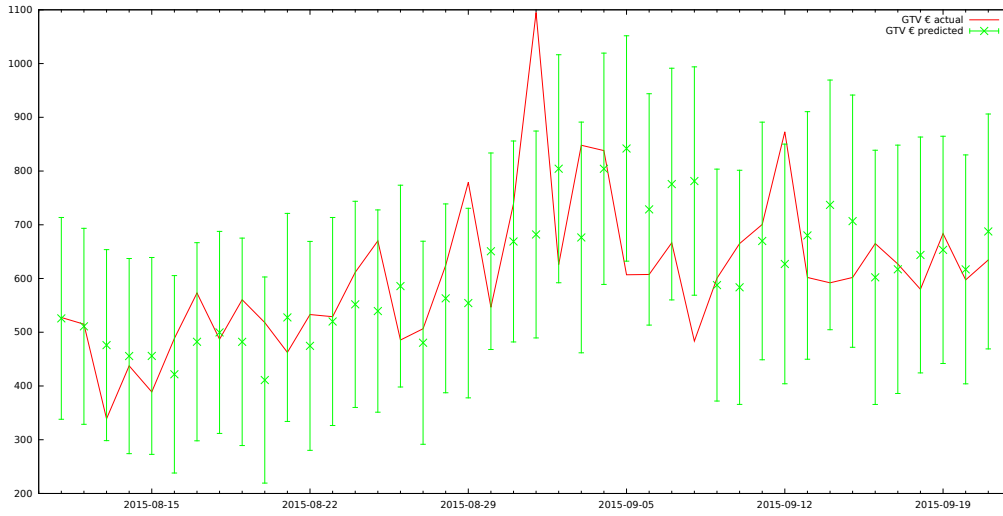


Figura 3.17: Falso positivo caso 1, predizione con training su 100 elementi

Il processo di prefiltraggio utilizza una scala stimata $\hat{\sigma}_t$ dell'errore r_t . Questa può essere calcolata applicando un qualsiasi stimatore di scala robusto applicato ai valori di r_t . Un possibile stimatore di scala robusto è la deviazione media assoluta (MAD), così definita:

$$\hat{\sigma}_t = \text{MAD}(r_s) = \text{med}_{1 \leq s \leq t} \left| r_s - \text{med}_{1 \leq s \leq t} r_s \right| \quad (3.9)$$

Questa assume che la scala rimanga pressoché costante ed una possibilità per aggiornare lentamente il valore di $\hat{\sigma}_t$ è quella di applicare la formula:

$$\hat{\sigma}_t^2 = \delta \rho \left(\frac{r_t}{\hat{\sigma}_{t-1}} \right) \hat{\sigma}_{t-1}^2 + (1 - \delta) \hat{\sigma}_{t-1}^2 \quad (3.10)$$

ρ è la funzione così definita:

$$\rho(x) = \begin{cases} c_k (1 - (1 - (\frac{x}{k})^2)^3) & \text{se } |x| \leq k \\ c_k & \text{altrimenti} \end{cases} \quad (3.11)$$

c_k è una costante necessaria a mantenere la coerenza del parametro di scala per una distribuzione normale dell'errore di distribuzione. Quando $k = 2$ il valore di c_k è 2.52.

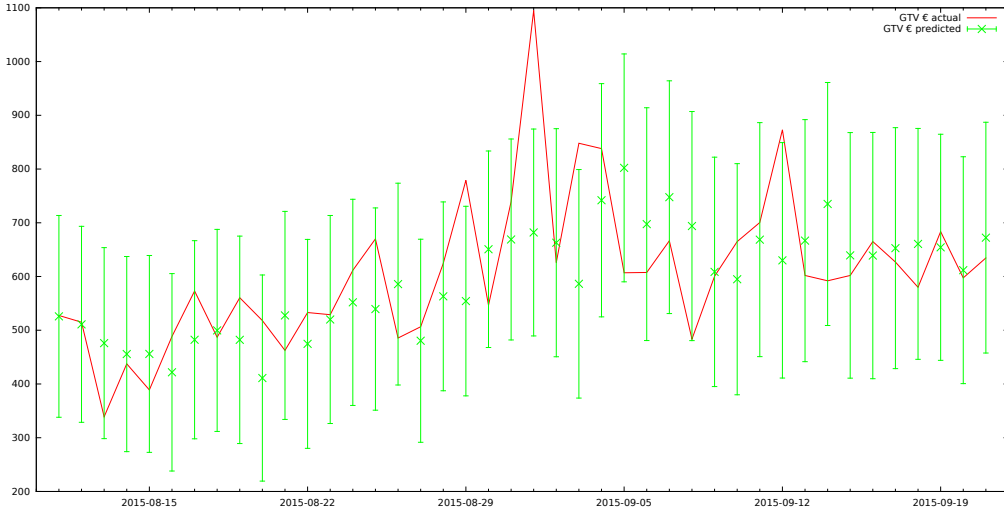


Figura 3.18: Falso positivo caso 1, predizione con Holt-Winters supervisionato e training su 100 elementi

Sostituendo l'equazione 3.7 all'interno di 3.6 si ottiene la versione robusta della tecnica Holt-Winters:

$$a_t = \alpha \left(\psi \left(\frac{y_t - (a_{t-1} + F_{t-1} + S_{t-s})}{\hat{\sigma}_t} \right) \hat{\sigma}_t + S_{t-s} \right) + a_{t-1} + F_{t-1} \quad (3.12)$$

$$F_t = \beta(a_t - a_{t-1}) + (1 - \beta)F_{t-1} \quad (3.13)$$

$$S_t = \gamma \left(\psi \left(\frac{y_t - (a_{t-1} + F_{t-1} + S_{t-s})}{\hat{\sigma}_t} \right) \hat{\sigma}_t + a_{t-1} + F_{t-1} \right) + S_{t-s} \quad (3.14)$$

Un ulteriore passo per migliorare le prestazioni dell'algoritmo è quella di utilizzare, per l'algoritmo di minimizzazione dell'errore una metrica che sia robusta agli outlier. La metrica SSE normalmente utilizzata con la tecnica Holt-Winters tradizionale non soddisfa questa proprietà in quanto anche un solo outlier può contribuire notevolmente, grazie anche al termine al quadrato ad innalzare il valore dell'errore. Per questo, basandoci su di uno stimatore τ di scala definiamo:

$$\tau^2(r_1, \dots, r_T) = S_T^2 \frac{1}{T} \sum_{t=1}^T \rho \left(\frac{r_t}{S_T} \right) \quad (3.15)$$

Dove $S_T = 1.48 \text{ Med}_t|r_t|$ e T è la lunghezza del periodo di training (esclusa la prima stagione di startup). La robustezza di τ^2 è dovuta al fatto che riduce il contributo degli errori di predizione elevati ed utilizza la funzione limitata ρ già definita in 3.11.

3.6.1 Confronto con la versione tradizionale

Caso	MSPE	
	HW	RHW
Vero positivo 1	22214	20658
Vero positivo 2	388379	219351
Falso positivo 1	15951	17421
Falso positivo 2	3718	4265
Falso negativo 1	15732	13985
Falso negativo 2	24120	25598

Tabella 3.2: Confronto dell'errore di predizione 100 giorni di training

Al fine di comparare le prestazioni della tecnica Holt-Winters (HW) con la tecnica Robust Holt-Winters (RHW) abbiamo misurato su di un campione di 100 previsioni e con 100 giorni di training, l'errore MSPE per ognuno degli esempi già illustrati nella precedente tabella.

Vediamo come l'errore sia migliore nei casi di vero positivo 1 e 2 e nel falso negativo 1, mentre peggiora nei casi di falso positivo 1 e 2 e nel falso negativo 2.

Tra gli esempi sopra riportati selezioniamo un vero positivo e un falso positivo al fine di analizzare qualitativamente l'andamento della predizione. In figura 3.19 riproponiamo la serie storica di figura 3.11 vediamo come il miglioramento dell'errore non sia facilmente apprezzabile dal punto di vista qualitativo. In figura 3.20 abbiamo invece il caso di falso positivo N° 2 di figura 2.9 anche in questo caso il leggero peggioramento di HW rispetto a RHW è difficilmente apprezzabile.

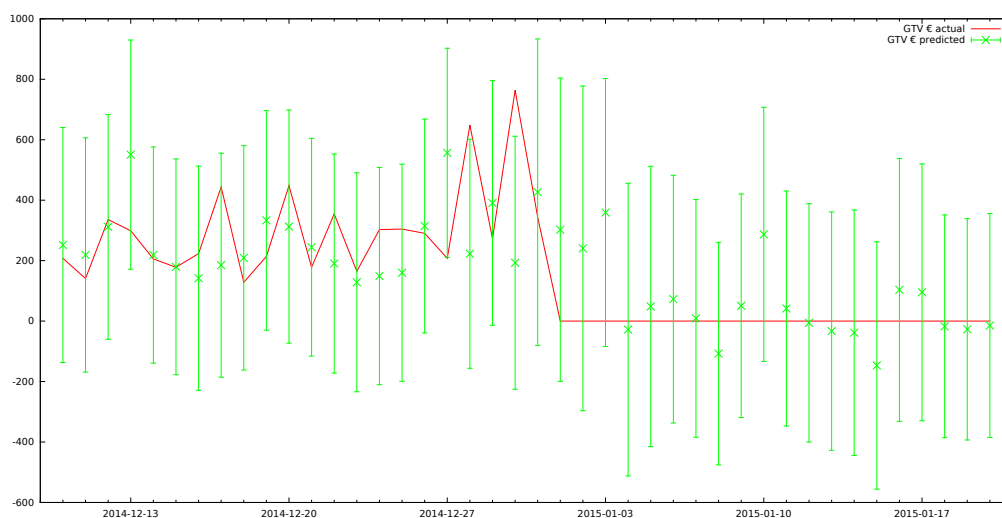


Figura 3.19: vero positivo con blocco del merchant (caso 1) RHW

3.6.2 Implementazione

Dato che la procedura predefinita HoltWinters non è modificabile in quanto parte del sistema R si è provveduto a creare un nuovo pacchetto installabile, effettuando il porting da C a C++ (quest'ultimo di più facile integrazione grazie all'estensione Rcpp [Edd13]).

La validazione del porting è stata effettuata utilizzando i dataset di esempio, verificando che tutti i dati numerici coincidessero con l'implementazione originale.

Infine la parte più corposa dello sviluppo è stata relativa all'implementazione della variante robusta di Holt-Winters.

Nell'appendice A viene riportato tutto il codice sorgente R e C++ scritto durante questa attività di tesi, oltre a quello del sistema esistente e della libreria di sistema HoltWinters. Il codice dell'infrastruttura di reportistica è omesso per ragioni di riservatezza.

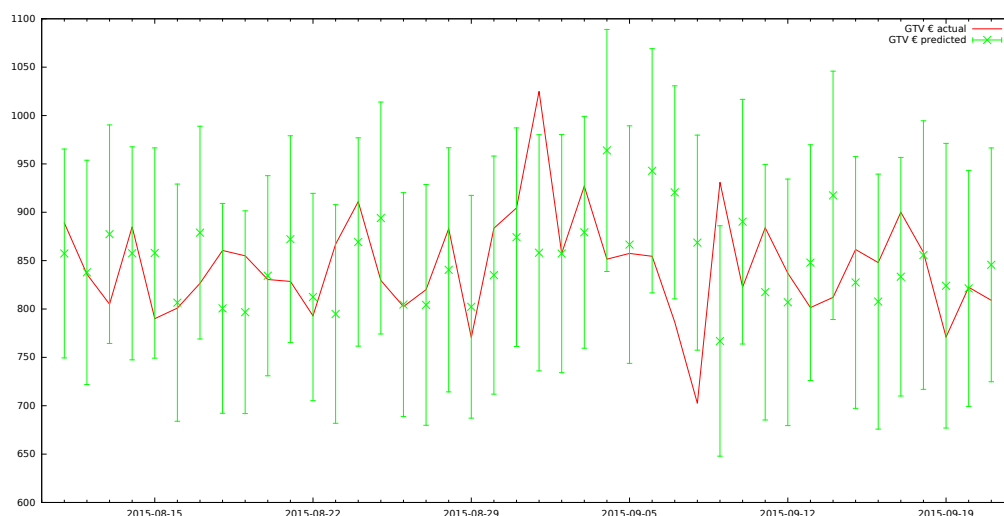


Figura 3.20: falso positivo del 01/09 dovuto ad un picco di attivazioni (caso 2) RHW

3.7 Conclusioni

In questo capitolo abbiamo analizzato come migliorare le performance dell'algoritmo, aumentando dapprima la lunghezza del periodo di training e successivamente valutando le ulteriori dimensioni di aggregazione, studiato una modalità supervisionata per l'eliminazione degli outlier ed infine applicata una variante robusta della tecnica Holt-Winters.

Abbiamo visto come l'allungamento del periodo di training migliori notevolmente le prestazioni dell'algoritmo con un buon compromesso quando la sua durata è di 100 giorni.

L'aggiunta di ulteriori dimensioni di aggregazione comporta una diminuzione dei costi operativi per l'indagine a fronte di un allarme attivato, in particolare per quanto riguarda la distinzione per business model.

Risulta inoltre particolarmente efficace la tecnica di esclusione supervisionata degli outlier, cosa ancora più evidente con periodi di training brevi (30 giorni).

Il passaggio da Holt-Winters a Robust Holt-Winters ha dato dei miglioramenti nei dati di veri positivo con outlier, mentre ha dimostrato un peggior-

ramento con le altre tipologie di dato, probabilmente per via di un filtraggio che *taglia* parzialmente picchi non anomali. Per questo motivo un possibile sviluppo futuro potrebbe consistere nella ricerca di un modo più efficace per il calcolo dei valori di $\bar{\sigma}_t$.

L'aggiunta di livelli maggiori di dettaglio e utilizzo di tecniche robuste possono essere opportuni su di un sottoinsieme di merchant con volumi di fatturato importante.

In conclusione, l'utilizzo combinato dell'esclusione supervisionata e un aumento a 100 giorni del periodo di training sono le migliori che danno un maggior beneficio, portando il funzionamento del sistema ad un livello più che soddisfacente.

Capitolo 4

Sviluppi futuri

In questo capitolo valuteremo eventuali sviluppi futuri che potrebbero aumentare notevolmente la correttezza del sistema esistente.

Nel capitolo 2 avevamo dimostrato come la mancanza della modellazione della stagionalità multipla fosse un problema potenzialmente maggiore del mancato smorzamento degli outlier. L'implementazione del sistema di rilevamento delle frodi basato sull'algoritmo Robust Holt-Winters infatti non aveva fornito risultati nettamente migliori rispetto alla versione basata sull'algoritmo Holt-Winters. Di conseguenza il principale sviluppo futuro che andremo a proporre in questo capitolo riguarda la modellazione delle componenti multiple di stagionalità.

4.1 Componenti multiple di stagionalità

Una possibile estensione della tecnica Holt-Winters è quella che porta a considerare stagionalità multiple [Tay03, GKO⁺08].

In particolare andremo a riassumere l'articolo [GKO⁺08]. Per comprendere la tecnica proposta è necessario introdurre il formalismo utilizzato nell'articolo, il quale è leggermente diverso da quello utilizzato precedentemente nel capitolo 2.

Le differenze tra i formalismi non sono molte e possono essere riassunte nella seguente tabella:

$$y_t = b_1 + b_2 t + S_t + \varepsilon_t \quad \equiv \quad y_t = l_{t-1} + b_{t-1} + S_{t-m} + \varepsilon_t, \quad (4.1a)$$

$$a_t = \alpha(y_t - S_{t-s}) + (1 - \alpha)(a_{t-1} + F_{t-1}) \quad \equiv \quad l_t = l_{t-1} + b_{t-1} + \alpha \varepsilon_t, \quad (4.1b)$$

$$F_t = \beta(a_t - a_{t-1}) + (1 - \beta)F_{t-1} \quad \equiv \quad b_t = b_{t-1} + \beta \varepsilon_t, \quad (4.1c)$$

$$S_t = \gamma(y_t - a_t) + (1 - \gamma)S_{t-s} \quad \equiv \quad S_t = S_{t-m} + \gamma_w \varepsilon_t, \quad (4.1d)$$

Nella colonna di sinistra abbiamo le equazioni del modello illustrato in sezione 2.3.3, mentre nella colonna di destra abbiamo il modello utilizzato in [GKO⁺08].

Ricordiamo l'assunzione $\varepsilon_t \sim \text{NID}(0, \sigma^2)$ ovvero che l'errore di predizione abbia una distribuzione normale. Chiameremo questo modello $HW(m)$ e definiremo così il ciclo stagionale:

$$\mathbf{c}_t = (S_t, S_{t-1}, \dots, S_{t-m+1})'. \quad (4.2)$$

La tecnica HW aggiorna ognuno dei parametri di stagionalità una sola volta per ogni ciclo stagionale.

Quindi, nel caso di serie a granularità oraria potremmo avere un modello $HW(24)$ con un ciclo di lunghezza 24 (giornaliero); in questo caso ognuno degli elementi di stagionalità sarebbe aggiornato ogni 24 ore. Un'altra possibilità è il modello $HW(168)$ nel quale la lunghezza del ciclo è settimanale (24 ore per 7 giorni); in questo caso, nonostante possa emergere un pattern giornaliero all'interno della settimana ognuno dei 168 elementi di stagionalità viene aggiornato solo settimanalmente. Inoltre la stessa costante γ_w viene utilizzata per ognuno degli m termini di stagionalità.

Di seguito andremo ad analizzare come sia possibile rilassare questi vincoli utilizzando il modello di stagionalità multipla (MS).

4.1.1 Stagionalità doppia

[Tay03] introduce una variante della tecnica HW che prevede una stagionalità multipla, siano m_1 e m_2 la lunghezza dei due cicli di stagionalità si

definisce il modello come segue:

$$y_t = l_{t-1} + b_{t-1} + S_{t-m_1}^{(1)} + S_{t-m_2}^{(2)} + \varepsilon_t, \quad (4.3a)$$

$$l_t = l_{t-1} + b_{t-1} + \alpha \varepsilon_t, \quad (4.3b)$$

$$b_t = b_{t-1} + \beta \varepsilon_t, \quad (4.3c)$$

$$S_t^{(1)} = S_{t-m_1} + \gamma_{d_1} \varepsilon_t, \quad (4.3d)$$

$$S_t^{(2)} = S_{t-m_2} + \gamma_{d_2} \varepsilon_t, \quad (4.3e)$$

Chiamiamo questo modello $DS(m_1, m_2)$ e definiamo così i due cicli di stagionalità:

$$\mathbf{c}_t^{(1)} = \left(S_t^{(1)}, S_{t-1}^{(1)}, \dots, S_{t-m_1+1}^{(1)} \right) \quad (4.4)$$

$$\mathbf{c}_t^{(2)} = \left(S_t^{(2)}, S_{t-1}^{(2)}, \dots, S_{t-m_2+1}^{(2)} \right) \quad (4.5)$$

In questo caso sarà necessario calcolare $m_1 + m_2 + 2$ valori iniziali per il modello.

Abbiamo quindi m_2 termini di stagionalità nel ciclo lungo che vengono aggiornati ogni m_2 unità di tempo. Inoltre abbiamo m_1 termini aggiuntivi di stagionalità nel ciclo più breve che sono aggiornati ogni m_1 unità di tempo. Nel modello $DS(m_1, m_2)$ non è richiesto che m_1 sia divisore di m_2 , tuttavia se questo requisito è soddisfatto e $k = \frac{m_2}{m_1}$, allora abbiamo k cicli brevi all'interno di ogni ciclo lungo.

Per i dati a granularità oraria avremmo 168 termini aggiornati una volta per ogni ciclo di 168 intervalli di tempo e altri 24 termini di stagionalità che sono aggiornati una volta ogni 24 unità di tempo. Per il ciclo lungo, lo stesso coefficiente di smussamento γ_{d_2} viene utilizzato per ognuno dei 168 termini, e allo stesso modo γ_{d_1} viene utilizzata per ogni elemento del ciclo più corto, di 24 unità di tempo. Nel modello a stagionalità multipla (MS) che andremo ad illustrare queste condizioni possono essere rilassate.

Andiamo ora ad utilizzare delle variabili di comodo per esprimere il modello $HW(m_2)$ in due nuove forme quando $k = \frac{m_2}{m_1}$. In questo modo rendiamo il modello MS e i suoi casi particolari più facili da comprendere. Come prima cosa suddividiamo il ciclo c_0 di $HW(m_2)$ in k sottocicli come segue:

$$\begin{aligned}
c_{i,0} &= (S_{i,0}, S_{i,-1}, \dots, S_{i,-m_1+1})' \\
&= (S_{-m_1(k-i)}, S_{-m_1(k-i)-1}, \dots, S_{-m_1(k-i)-m_1+1})'
\end{aligned} \tag{4.6}$$

Per esempio, con i dati a risoluzione oraria, potremmo dividere il ciclo settimanale di lunghezza 168 in $k = 7$ sottocicli giornalieri di lunghezza $m_1 = 24$. Per ogni intervallo di tempo t , c_{it} contiene i valori correnti delle m_1 componenti di stagionalità per il ciclo i (es. il giorno i) ed è definita da:

$$c_{it} = (S_{i,t}, S_{i,t-1}, \dots, S_{i,t-m_1+1}) \quad i = 1, \dots, k \tag{4.7}$$

Successivamente definiamo un insieme di variabili di comodo che indicano in quale sottociclo ci troviamo nell'intervallo di tempo t . Per esempio, utilizzando dati a granularità oraria queste variabili di comodo indicheranno il ciclo giornaliero a cui l'intervallo di tempo appartiene. Queste variabili sono definite come segue:

$$x_{it} = \begin{cases} 1 & \text{se l'intervallo } t \text{ fa parte dell}'i\text{-esimo sottociclo} \\ 0 & \text{altrimenti} \end{cases} \tag{4.8}$$

Quindi il modello $HW(m_2)$ può essere scritto come segue:

$$y_t = l_{t-1} + b_{t-1} + \sum_{i=1}^k x_{it} S_{i,t-m_1} + \varepsilon_t, \tag{4.9a}$$

$$l_t = l_{t-1} + b_{t-1} + \alpha \varepsilon_t, \tag{4.9b}$$

$$b_t = b_{t-1} + \beta \varepsilon_t, \tag{4.9c}$$

$$S_{it} = S_{i,t-m_1} + \gamma_w x_{it} \varepsilon_t \quad (i = 1, \dots, k). \tag{4.9d}$$

Lo scopo di x_{it} è di fare in modo che gli $m_2 (= k \times m_1)$ elementi di stagionalità vengano aggiornati esattamente ogni m_2 intervalli di tempo. L'equazione 4.9d può essere scritta come caso particolare del modello MS che andremo a vedere come segue:

$$S_{it} = S_{i,t-m_1} + \left(\sum_{j=1}^k \gamma_{ij} x_{jt} \right) \varepsilon_t \quad (i = 1, \dots, k), \quad (4.10)$$

dove

$$\gamma_{ij} = \begin{cases} \gamma_w & \text{se } i = j \\ 0 & \text{altrimenti} \end{cases} \quad (4.11)$$

4.1.2 Modello a stagionalità multipla

Un obiettivo fondamentale del modello per le stagionalità multiple è quello di permettere che gli elementi di stagionalità che rappresentano un ciclo vengano aggiornati più di una volta all'interno del periodo dello stesso. Questo può essere ottenuto in due modi con il nostro modello. Cominceremo, come abbiamo già fatto per il modello $HW(m_2)$ nella precedente sezione, dividendo il ciclo di lunghezza m_2 in k cicli più corti di lunghezza m_1 .

Successivamente usiamo una matrice di parametri di smussamento in modo da permettere che i termini stagionali di un sottociclo possano essere aggiornati durante il tempo di un altro sottociclo. Per esempio i termini di stagionalità del lunedì possono essere aggiornati anche il martedì. A volte questo obiettivo può essere raggiunto combinando i sottocicli con lo stesso *pattern* di stagionalità in un unico sottociclo. Quest'ultimo approccio ha il vantaggio di ridurre il numero di valori iniziali necessari.

Un aggiornamento più frequente dei valori potrebbe migliorare la qualità delle predizioni, in particolare quando le osservazioni effettuate m_1 unità di tempo nel passato sono più importanti di quelle effettuate m_2 unità di tempo nel passato. Con questo modello è anche possibile avere differenti parametri di smoothing per i differenti sottocicli (es. diversi giorni della settimana).

L'esistenza di sottocicli comuni è la chiave per la riduzione dei valori iniziali rispetto a quelli richiesti dalle tecniche Holt-Winters con stagionalità singola e doppia.

Come già descritto nella sezione 4.1.1 è possibile suddividere un ciclo lungo in $k = \frac{m_2}{m_1}$ cicli più corti di lunghezza m_1 . Di questi possibili sottocicli

possono essere identificati $r \leq k$ cicli distinti. Per esempio, considerando l'esempio dove $m_1 = 24$ e $m_2 = 168$ nel caso di dati a granularità oraria. Supponendo che i giorni dal lunedì al venerdì abbiano lo stesso pattern di stagionalità possiamo utilizzare lo stesso sottociclo per questi 5 giorni. Allo stesso possiamo utilizzare lo stesso ciclo per sabato e domenica se questi sono simili; perciò possiamo ridurre il numero di sottocicli giornalieri da $k = 7$ ad $r = 2$.

Il numero di valori iniziali per le componenti di stagionalità scende così da 168 per la tecnica HW a 48 per questa nuova tecnica (per la tecnica DS il numero di valori iniziali richiesti era di 192).

Andiamo a definire, analogamente a come fatto in precedenza la seguente variabile di comodo:

$$x_{it} = \begin{cases} 1 & \text{se l'intervallo } t \text{ fa parte dell}'i\text{-esimo sottociclo} \\ 0 & \text{altrimenti} \end{cases} \quad (4.12)$$

Per ogni giorno considerato, solamente uno dei valori x_{it} varrà 1. Siano $\mathbf{x}_t = [x_{1t}, x_{2t}, x_{3t}, \dots, x_{rt}]'$ e $\mathbf{s}_t = [S_{1t}, S_{2t}, S_{3t}, \dots, S_{rt}]'$.

La formula generale del modello MS per $r \leq k = \frac{m_2}{m_1}$ è:

$$y_t = l_{t-1} + b_{t-1} + \sum_{i=1}^k x_{it} S_{i,t-m_1} + \varepsilon_t, \quad (4.13a)$$

$$l_t = l_{t-1} + b_{t-1} + \alpha \varepsilon_t, \quad (4.13b)$$

$$b_t = b_{t-1} + \beta \varepsilon_t, \quad (4.13c)$$

$$S_{it} = S_{i,t-m_1} + \left(\sum_{j=1}^k y_{ij} x_{jt} \right) \varepsilon_t \quad (i = 1, \dots, k), \quad (4.13d)$$

Dove $\varepsilon_t \sim \text{NID}(0, \sigma^2)$

Queste equazioni possono essere scritte anche in forma matriciale:

$$y_t = l_{t-1} + b_{t-1} + \varepsilon_t, \quad (4.14a)$$

$$l_t = l_{t-1} + b_{t-1} + \alpha \varepsilon_t, \quad (4.14b)$$

$$b_t = b_{t-1} + \beta \varepsilon_t, \quad (4.14c)$$

$$\mathbf{s}_t = \mathbf{s}_{t-m_1} + \mathbf{\Gamma} \mathbf{x}_t \quad (4.14d)$$

$$\hat{y}_t(1) = l_{t-1} + b_{t-1} + \mathbf{x}'_t \mathbf{s}_{t-m_1} \quad (4.14e)$$

Dove $\hat{y}_t(1)$ è la predizione del valore y_t fatta con i dati fino a y_{t-1} , (in altri articoli espressa come $\hat{y}_{t|t-1}$).

$\mathbf{\Gamma}$ è la matrice di smussamento della stagionalità, che contiene i parametri di smussamento per ognuno dei cicli. Il parametro γ_{ii} è utilizzato per aggiornare i termini di stagionalità negli intervalli che appartengono allo stesso sottociclo (es. giorni che hanno lo stesso pattern giornaliero). I parametri $\gamma_{ij}, i \neq j$ sono utilizzati per aggiornare i termini di stagionalità che appartengono ad un sottociclo differente.

Denomineremo questo modello $MS(r; m_1, m_2)$ e i cicli di stagionalità:

$$\mathbf{c}_{it} = (S_{i,t}, S_{i,t-1}, \dots, S_{i,t-m_1+1})' \quad i = 1, \dots, r \quad (4.15)$$

Mentre la rappresentazione 4.14 è utile nello sviluppo software, dove possono essere utilizzati ambienti ottimizzati per la manipolazione delle matrici, la rappresentazione 4.13 è utile per la comprensione del funzionamento del modello.

Per esempio, supponiamo di avere dei dati a granularità oraria con un pattern differente per ognuno dei 7 giorni della settimana (in questo modo otteniamo un pattern che si ripete ogni 168h).

Se il periodo t occorresse, ipotizziamo, nel giorno 3 della settimana, allora il ciclo \mathbf{c}_{3t} è in corso. Quindi, $x_{3t} = 1$ e $x_{jt} = 0 \forall j \neq 3$. Se $\gamma_{3,3} \neq 0$, allora la componente stagionale $S_{3,t}$ verrà aggiornata. Questa è la componente di stagionalità per il giorno della settimana corrispondente al tempo t . In aggiunta, ipotizzando $\gamma_{2,3} \neq 0$, la componente di stagionalità S_2 verrà aggiornata e corrisponde allo stesso orario del giorno ma nel giorno 2.

Inoltre, se i giorni 2 e 3 hanno lo stesso pattern, possiamo ridurre il numero di cicli di uno utilizzando il ciclo $c_{2,t}$ per rappresentare entrambi i giorni. Quindi se il tempo t fa parte del giorno 2 o 3, allora la variabile di comodo è definita per selezionare questo ciclo.

4.2 Conclusioni

In conclusione l'implementazione del modello con stagionalità multiple permetterebbe una riduzione dell'errore di predizione MSPE, infatti essa permetterebbe all'algoritmo di tenere in considerazione più di un pattern di stagionalità evitando quindi di cadere in falsi positivi evitabili.

Rimangono aperte le seguenti problematiche:

- gestione della stagionalità mensile (dato che la lunghezza dei mesi non è costante);
- gestione degli anni bisestili.

Appendice A

Codice sorgente

In questa appendice trovate il codice sorgente della nostra implementazione in R e C++ oltre ai riferimenti per ottenere i sorgenti della libreria originale.

A.1 Predizione mediante Holt-Winters

```
1 args <- commandArgs(trailingOnly = TRUE)
2 frequency <- as.numeric(args[1])
3 numberOfPredictions <- 1
4 fileWithData <- file("stdin")
5
6 amountsWithDates <- read.csv(fileWithData, header=FALSE)
7 numberOfDays <- length(amountsWithDates[,1])
8
9 historyBeforePrediction <- 1:numberOfDays-1
10 amountsTS <- ts(amountsWithDates[historyBeforePrediction,2],
11                frequency = frequency)
12 model <- HoltWinters(amountsTS)
13 predictedFutureAmounts <- predict(model, numberOfPredictions,
14                                   prediction.interval=TRUE)
15
16 when <- as.vector(amountsWithDates[numberOfDays,1])
```

```
17 predicted <- as.numeric(predictedFutureAmounts[1, 'fit'])
18 measured <- as.numeric(amountsWithDates[numberOfDays, 2])
19 lower <- as.numeric(predictedFutureAmounts[1, 'lwr'])
20 upper <- as.numeric(predictedFutureAmounts[1, 'upr'])
21 # prediction distance
22 pd <- measured - predicted
23 # boundary distance
24 bd <- 0
25 if (measured < lower || measured > upper) {
26   bd <- min(abs(measured - upper), abs(measured - lower)) *
27     sign(measured - predicted)
28 }
29
30 alpha = as.numeric(model[ 'alpha '])
31 beta = as.numeric(model[ 'beta '])
32 gamma = as.numeric(model[ 'gamma '])
33 SSE = as.numeric(model[ 'SSE '])
34 MSPE = SSE / (numberOfDays - frequency - 1)
35
36 cat(when, "\n")
37 cat(measured, "\n")
38 cat(round(predicted, digits=2), "\n")
39 cat(round(upper, digits=2), "\n")
40 cat(round(lower, digits=2), "\n")
41 cat(round(pd, digits=2), "\n")
42 cat(round(bd, digits=2), "\n")
43 cat(alpha, "\n")
44 cat(beta, "\n")
45 cat(gamma, "\n")
46 cat(SSE, "\n")
47 cat(MSPE, "\n")
```

A.2 Procedura HoltWinters standard di R

Per la parte in R si visiti l'indirizzo: <https://svn.r-project.org/R/tags/R-2-14-1/src/library/stats/R/HoltWinters.R>, per la parte in C: <https://svn.r-project.org/R/tags/R-2-14-1/src/library/stats/src/HoltWinters.c>

A.3 Predizione mediante Supervised Holt-Winters

```
1 library( RobustHoltWinters )
2
3 args <- commandArgs( trailingOnly = TRUE )
4 frequency <- as.numeric( args [ 1 ] )
5 numberOfPredictions <- 1
6 fileWithData <- file( "stdin" )
7
8 amountsWithDates <- read.csv( fileWithData , header=FALSE )
9 numberOfDays <- length( amountsWithDates [ , 1 ] )
10
11 historyBeforePrediction <- 1:numberOfDays-1
12 amountsTS <- ts( amountsWithDates [ historyBeforePrediction , 2 ] ,
13               frequency = frequency )
14 anomalous <- as.integer( amountsWithDates [ historyBeforePrediction , 3 ] )
15 model <- SupervisedHoltWinters( amountsTS , outlier = anomalous )
16 predictedFutureAmounts <- predict( model , numberOfPredictions ,
17                                   prediction.interval=TRUE )
18
19 when <- as.vector( amountsWithDates [ numberOfDays , 1 ] )
20 predicted <- as.numeric( predictedFutureAmounts [ 1 , 'fit' ] )
21 measured <- as.numeric( amountsWithDates [ numberOfDays , 2 ] )
22 lower <- as.numeric( predictedFutureAmounts [ 1 , 'lwr' ] )
23 upper <- as.numeric( predictedFutureAmounts [ 1 , 'upr' ] )
24 # prediction distance
25 pd <- measured - predicted
```

```

26 # boundary distance
27 bd <- 0
28 if (measured < lower || measured > upper) {
29   bd <- min(abs(measured - upper), abs(measured - lower))
30   * sign(measured - predicted)
31 }
32
33 alpha = as.numeric(model[ 'alpha' ])
34 beta = as.numeric(model[ 'beta' ])
35 gamma = as.numeric(model[ 'gamma' ])
36 SSE = as.numeric(model[ 'SSE' ])
37 MSPE = SSE / (numberOfDays - frequency - 1)
38
39 cat(when, "\n")
40 cat(measured, "\n")
41 cat(round(predicted, digits=2), "\n")
42 cat(round(upper, digits=2), "\n")
43 cat(round(lower, digits=2), "\n")
44 cat(round(pd, digits=2), "\n")
45 cat(round(bd, digits=2), "\n")
46 cat(alpha, "\n")
47 cat(beta, "\n")
48 cat(gamma, "\n")
49 cat(SSE, "\n")
50 cat(MSPE, "\n")

```

A.4 Procedura Supervised Holt-Winters

A.4.1 Parte R

```

1
2 # File src/library/stats/R/HoltWinters.R
3 # Part of the R package, http://www.R-project.org
4 #
5 # This program is free software; you can redistribute it and/or modify

```



```
6 # it under the terms of the GNU General Public License as published by
7 # the Free Software Foundation; either version 2 of the License, or
8 # (at your option) any later version.
9 #
10 # This program is distributed in the hope that it will be useful,
11 # but WITHOUT ANY WARRANTY; without even the implied warranty of
12 # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 # GNU General Public License for more details.
14 #
15 # A copy of the GNU General Public License is available at
16 # http://www.r-project.org/Licenses/
17
18 # Originally contributed by David Meyer
19
20 SupervisedHoltWinters <-
21 function (x, seasonal = c("additive", "multiplicative"), outlier = F)
22 {
23   # starting values for optim
24   optim.start = c(alpha = 0.3, beta = 0.1, gamma = 0.1)
25   optim.control = list()
26   start.periods = 2
27
28   x <- as.ts(x)
29   seasonal <- match.arg(seasonal)
30   f <- frequency(x)
31
32   ## initialization
33   ## seasonal Holt-Winters
34   start.time <- f + 1
35   wind <- start.periods * f
36
37   ## decompose series
38   st <- decompose(ts(x[1L:wind], start = start(x), frequency = f),
39                 seasonal)
40
```

```

41  ## level & intercept
42  dat <- na.omit(st$trend)
43  m   <- lm(dat ~ seq_along(dat))
44
45  l.start <- as.vector(coef(m)[1L])
46  b.start <- as.vector(coef(m)[2L])
47  s.start <- st$figure
48
49  ## Call to filtering loop
50  lenx <- as.integer(length(x))
51  if (is.na(lenx)) stop("invalid length(x)")
52
53  len <- lenx - start.time + 1
54  st <- function (residuals) {
55      return (as.double(1.48 * median(abs(residuals))))
56  }
57  hw <- function (alpha, beta, gamma) {
58      return (SupervisedHoltWintersCpp(
59          x,
60          outlier,
61          as.double(alpha),
62          as.double(beta),
63          as.double(gamma),
64          as.integer(start.time),
65          as.integer(f),
66          l.start,
67          b.start,
68          s.start
69      ))
70  }
71
72  ## --> optimize alpha, beta, and gamma
73  error <- function (p) hw(p[1L], p[2L], p[3L])$SSE
74  sol   <- optim(optim.start, error, method = "L-BFGS-B",
75              lower = c(0, 0, 0), upper = c(1, 1, 1),

```

```
76         control = optim.control)
77     if(sol$convergence == any(sol$par < 0 | sol$par > 1)) {
78         if (sol$convergence > 50) {
79             warning(gettextf("optimization difficulties: %s",
80                             sol$message), domain = NA)
81         } else stop("optimization failure")
82     }
83     alpha <- sol$par[1L]
84     beta  <- sol$par[2L]
85     gamma <- sol$par[3L]
86
87     ## get (final) results
88     final.fit <- hw(alpha, beta, gamma)
89
90     # return fitted values and estimated coefficients
91     # along with parameters used
92     fitted <- ts(cbind(xhat   = final.fit$level[-len-1],
93                       level  = final.fit$level[-len-1],
94                       trend   = final.fit$trend[-len-1],
95                       season  = final.fit$seasonal[1L:len]),
96                start = start(lag(x, k = 1 - start.time)),
97                frequency = frequency(x)
98                )
99     fitted[,1] <- fitted[,1] + fitted[, "trend"]
100    fitted[,1] <- if (seasonal == "multiplicative")
101        fitted[,1] * fitted[, "season"]
102    else
103        fitted[,1] + fitted[, "season"]
104
105    structure(list(fitted   = fitted,
106                  x       = x,
107                  alpha   = alpha,
108                  beta    = beta,
109                  gamma   = gamma,
110                  coefficients = c(a = final.fit$level[len + 1],
```

```

111             b = final.fit$trend[len + 1],
112             s = final.fit$seasonal[len + 1L:f]),
113         seasonal = seasonal,
114         SSE      = final.fit$SSE,
115         tau2     = final.fit$tau2,
116         call     = match.call()
117     ),
118     class = "HoltWinters"
119 )
120 }

```

A.4.2 Parte C++

```

#include <Rcpp.h>
#include <assert.h>
#include <cmath>

using namespace Rcpp;

// [[Rcpp::export]]
List SupervisedHoltWintersCpp(
    NumericVector x,
    IntegerVector outlier,
    const double alpha,
    const double beta,
    const double gamma,
    int startTime,
    int frequency,
    double levelInitial,
    double trendInitial,
    NumericVector seasonInitial
) {
    const int xLength = x.length();
    const int smoothedLength = xLength - startTime + 1;
    const int seasonLength = frequency;

```

```
NumericVector level(smoothedLength + 1);
NumericVector trend(smoothedLength + 1);
NumericVector season(smoothedLength + seasonLength);
NumericVector residuals(smoothedLength);

/* copy start values to the beginning of the vectors */
level[0] = levelInitial;
trend[0] = trendInitial;

for (int i = 0; i < seasonLength; ++i) {
    season[i] = seasonInitial[i];
}

double SSE = 0.0;

for (int t = startTime - 1; t < xLength; t++) {
    /* indices for period i */
    const int currentIndex = t - startTime + 2;
    const int currentResidualIndex = currentIndex - 1;
    const int previousIndex = currentIndex - 1;
    const int currentSeasonalIndex = currentIndex + seasonLength - 1;
    const double xAtT = x[t];

    /* forecast *for* period i */
    // xhat is the 1 step ahead prediction
    const double seasonalPrevious =
        season[currentSeasonalIndex - seasonLength];
    const double xhat =
        level[previousIndex] + trend[previousIndex] + seasonalPrevious;

    /* Sum of Squared Errors */
    double residual = residuals[currentResidualIndex] = xAtT - xhat;
    double xFiteredAtT = xAtT;
```

```

    if (outlier[t]) {
        residual = 0;
        xFiteredAtT = xhat;
    }

    SSE += pow(residual, 2);

    /* estimate of level *in* period t */
    level[currentIndex] = alpha * (xFiteredAtT - seasonalPrevious)
        + (1 - alpha) * (level[previousIndex] + trend[previousIndex]);

    /* estimate of trend *in* period t */
    trend[currentIndex] =
        beta * (level[currentIndex] - level[previousIndex])
        + (1 - beta) * trend[previousIndex];

    /* estimate of seasonal component *in* period t */
    season[currentSeasonalIndex] =
        gamma * (xFiteredAtT - level[currentIndex])
        + (1 - gamma) * seasonalPrevious;
}

List output;

output["SSE"] = SSE;
output["level"] = level;
output["trend"] = trend;
output["seasonal"] = season;

return output;
}

```

A.5 Predizione mediante Robust Holt-Winters

```
1 library(RobustHoltWinters)
```

```
2
3 args <- commandArgs(trailingOnly = TRUE)
4 frequency <- as.numeric(args[1])
5 numberOfPredictions <- 1
6 fileWithData <- file("stdin")
7
8 amountsWithDates <- read.csv(fileWithData, header=FALSE)
9 numberOfDays <- length(amountsWithDates[,1])
10
11 historyBeforePrediction <- 1:numberOfDays-1
12 amountsTS <- ts(amountsWithDates[historyBeforePrediction,2],
13                 frequency = frequency)
14 model <- RobustHoltWinters(amountsTS)
15 predictedFutureAmounts <- predict(model, numberOfPredictions,
16                                   prediction.interval=TRUE)
17
18 when <- as.vector(amountsWithDates[numberOfDays,1])
19 predicted <- as.numeric(predictedFutureAmounts[1,'fit'])
20 measured <- as.numeric(amountsWithDates[numberOfDays,2])
21 lower <- as.numeric(predictedFutureAmounts[1,'lwr'])
22 upper <- as.numeric(predictedFutureAmounts[1,'upr'])
23 # prediction distance
24 pd <- measured - predicted
25 # boundary distance
26 bd <- 0
27 if (measured < lower || measured > upper) {
28   bd <- min(abs(measured - upper), abs(measured - lower)) *
29     sign(measured - predicted)
30 }
31
32 alpha = as.numeric(model['alpha'])
33 beta = as.numeric(model['beta'])
34 gamma = as.numeric(model['gamma'])
35 SSE = as.numeric(model['SSE'])
36 MSPE = SSE / (numberOfDays - frequency - 1)
```

```
37
38 cat(when, "\n")
39 cat(measured, "\n")
40 cat(round(predicted, digits=2), "\n")
41 cat(round(upper, digits=2), "\n")
42 cat(round(lower, digits=2), "\n")
43 cat(round(pd, digits=2), "\n")
44 cat(round(bd, digits=2), "\n")
45 cat(alpha, "\n")
46 cat(beta, "\n")
47 cat(gamma, "\n")
48 cat(SSE, "\n")
49 cat(MSPE, "\n")
```

A.6 Procedura Robust Holt-Winters

A.6.1 Parte R

```
1 # This program is free software; you can redistribute it and/or modify
2 # it under the terms of the GNU General Public License as published by
3 # the Free Software Foundation; either version 3 of the License, or
4 # (at your option) any later version.
5 #
6 # This program is distributed in the hope that it will be useful,
7 # but WITHOUT ANY WARRANTY; without even the implied warranty of
8 # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
9 # GNU General Public License for more details.
10 #
11 # A copy of the GNU General Public License is available at
12 # http://www.r-project.org/licenses/
13
14 # Robust Holt-Winters based on R source code
15 # Originally contributed by David Meyer
16
17 RobustHoltWinters <-
```



```
18 function (x, seasonal = c("additive", "multiplicative"))
19 {
20     # starting values for optim
21     optim.start = c(alpha = 0.3, beta = 0.1, gamma = 0.1)
22     optim.control = list()
23     start.periods = 2
24
25     x <- as.ts(x)
26     seasonal <- match.arg(seasonal)
27     f <- frequency(x)
28
29     ## initialization
30     ## seasonal Holt-Winters
31     start.time <- f + 1
32     wind      <- start.periods * f
33
34     ## decompose series
35     st <- decompose(ts(x[1L:wind], start = start(x), frequency = f),
36                   seasonal)
37
38     ## level & intercept
39     dat <- na.omit(st$trend)
40     m   <- lm(dat ~ seq_along(dat))
41
42     l.start <- as.vector(coef(m)[1L])
43     b.start <- as.vector(coef(m)[2L])
44     s.start <- st$figure
45
46     ## Call to filtering loop
47     lenx <- as.integer(length(x))
48     if (is.na(lenx)) stop("invalid length(x)")
49
50     len <- lenx - start.time + 1
51     st <- function (residuals) {
52         return (as.double(1.48 * median(abs(residuals))))
```

```

53   }
54   hw <- function (alpha, beta, gamma) {
55     return (RobustHoltWintersCpp(
56       x,
57       as.double(alpha),
58       as.double(beta),
59       as.double(gamma),
60       as.integer(start.time),
61       as.integer(f),
62       l.start,
63       b.start,
64       s.start,
65       mad(x[1:start.time - 1]),
66       st
67     ))
68   }
69
70   ## —> optimize alpha, beta, and gamma
71   error <- function (p) hw(p[1L], p[2L], p[3L])$tau2
72   sol <- optim(optim.start, error, method = "L-BFGS-B",
73             lower = c(0, 0, 0), upper = c(1, 1, 1),
74             control = optim.control)
75   if(sol$convergence || any(sol$par < 0 | sol$par > 1)) {
76     if (sol$convergence > 50) {
77       warning(gettextf("optimization difficulties: %s",
78                       sol$message), domain = NA)
79     } else stop("optimization failure")
80   }
81   alpha <- sol$par[1L]
82   beta <- sol$par[2L]
83   gamma <- sol$par[3L]
84
85   ## get (final) results
86   final.fit <- hw(alpha, beta, gamma)
87

```

```

88     # return fitted values and estimated coefficients
89     # along with parameters used
90     fitted <- ts(cbind(xhat   = final.fit$level[-len-1],
91                       level  = final.fit$level[-len-1],
92                       trend   = final.fit$trend[-len-1],
93                       season  = final.fit$seasonal[1L:len]),
94                 start = start(lag(x, k = 1 - start.time)),
95                 frequency = frequency(x)
96                 )
97     fitted[,1] <- fitted[,1] + fitted[, "trend"]
98     fitted[,1] <- if (seasonal == "multiplicative")
99         fitted[,1] * fitted[, "season"]
100    else
101        fitted[,1] + fitted[, "season"]
102
103    structure(list(fitted   = fitted ,
104                 x         = x,
105                 alpha     = alpha ,
106                 beta      = beta ,
107                 gamma     = gamma,
108                 coefficients = c(a = final.fit$level[len + 1],
109                                 b = final.fit$trend[len + 1],
110                                 s = final.fit$seasonal[len + 1L:f]),
111                 seasonal  = seasonal ,
112                 SSE       = final.fit$$SSE,
113                 tau2      = final.fit$tau2 ,
114                 call      = match.call()
115                 ),
116            class = "HoltWinters"
117            )
118 }

```

A.6.2 Parte C++

/*

```
* R robust Holt–Winters extension
* Copyright (C) 2015 Davide Bellettini
*
* based on R: A Computer Language for Statistical Data Analysis
* Copyright (C) 2003–7 The R Development Core Team
*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU General Public License as published by
* the Free Software Foundation; either version 3 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, a copy is available at
* http://www.gnu.org/licenses/gpl-3.0.txt
*/

#include <Rcpp.h>
#include <assert.h>
#include <cmath>

const double k = 2.0, ck = 2.52;

using namespace Rcpp;

double sgn(double x)
{
  if (x < 0) {
    return -1.0;
  }
}
```

```
    return 1.0;
}

/**
 * saturation function , returns x in the interval [-k, +k]
 * sign(x) * k otherwise
 */
// [[Rcpp::export]]
double psi (double x)
{
    if (std::abs(x) > k) {
        return sgn(x) * k;
    }

    return x;
}

/*
 * res is the residual (e.g. x - xhat) thus the difference
 * between actual value and 1 step ahead prediction
 */
bool shouldSaturate(double residual , double sigma)
{
    return (residual / sigma) > k || (residual / sigma) < -k;
}

/*
 * rho is a bounded loss function that is used to replace
 * pow(x, 2) to get robust
 *
 * returns double [0, ck]
 */
double rho (double x)
{
    if (x > k || x < -k) {
```

```

        return ck;
    }

    return ck * (1 - pow(1 - pow(x / k, 2), 3));
}

/**
 * sigma estimates the scale of error and it's needed for the
 * saturation, and takes sigma up to time t-1
 */
double updatesigma (double delta, double rt, double sigma)
{
    return sqrt(
        delta * rho(rt / sigma) * pow(sigma, 2)
        + (1 - delta) * pow(sigma, 2)
    );
}

double tau2(NumericVector residuals, double st)
{
    double sum = 0.0;
    int length = residuals.length();

    for (int i = 0; i < length; ++i) {
        sum += rho(residuals[i] / st);
    }

    return st * st * sum / length;
}

// [[Rcpp::export]]
List RobustHoltWintersCpp(
    NumericVector x,
    const double alpha,
    const double beta,

```

```
    const double gamma,
    int startTime,
    int frequency,
    double levelInitial,
    double trendInitial,
    NumericVector seasonInitial,
    double sigma,
    Function st
) {
    const int xLength = x.length();
    const int smoothedLength = xLength - startTime + 1;
    const int seasonLength = frequency;

    // sigma smoothing parameter (see the function updatesigma)
    double delta = 0.2;

    NumericVector level(smoothedLength + 1);
    NumericVector trend(smoothedLength + 1);
    NumericVector season(smoothedLength + seasonLength);
    NumericVector residuals(smoothedLength);

    /* copy start values to the beginning of the vectors */
    level[0] = levelInitial;
    trend[0] = trendInitial;

    for (int i = 0; i < seasonLength; ++i) {
        season[i] = seasonInitial[i];
    }

    double SSE = 0.0;

    for (int t = startTime - 1; t < xLength; t++) {
        /* indices for period i */
        const int currentIndex = t - startTime + 2;
        const int currentResidualIndex = currentIndex - 1;
```

```

const int previousIndex = currentIndex - 1;
const int currentSeasonalIndex = currentIndex + seasonLength - 1;
const double xAtT = x[t];

/* forecast *for* period i */
// xhat is the 1 step ahead prediction
const double seasonalPrevious =
    season[currentSeasonalIndex - seasonLength];
const double xhat =
    level[previousIndex] + trend[previousIndex] + seasonalPrevious;

/* Sum of Squared Errors */
double residual = residuals[currentResidualIndex] = xAtT - xhat;
double xFiteredAtT = xAtT;

if (shouldSaturate(residual, sigma)) {
    residual = sigma * psi(residual / sigma);
    xFiteredAtT = xhat + residual;
}

SSE += pow(residual, 2);

/* estimate of level *in* period t */
level[currentIndex] = alpha * (xFiteredAtT - seasonalPrevious)
    + (1 - alpha) * (level[previousIndex] + trend[previousIndex]);

/* estimate of trend *in* period t */
trend[currentIndex] =
    beta * (level[currentIndex] - level[previousIndex])
    + (1 - beta) * trend[previousIndex];

/* estimate of seasonal component *in* period t */
season[currentSeasonalIndex] =
    gamma * (xFiteredAtT - level[currentIndex])
    + (1 - gamma) * seasonalPrevious;

```



```
        sigma = updatesigma(delta , residual , sigma);
    }

    NumericVector stVector = st(residuals);

    List output;

    output["SSE"] = SSE;
    output["tau2"] = tau2(residuals , stVector[0]);
    output["level"] = level;
    output["trend"] = trend;
    output["seasonal"] = season;

    return output;
}
```


Conclusioni

Questa tesi ha cercato di evidenziare le caratteristiche del mercato del *carrier billing* concentrandosi sulle problematiche inerenti l'individuazione delle frodi.

Viene inizialmente descritto il sistema attuale, basato sull'analisi delle serie storiche relative ai volumi giornalieri di pagamento mediante la tecnica Holt-Winters in ambiente R.

Successivamente vengono individuate e messe in atto le seguenti migliorie:

- aumento della lunghezza del periodo di training;
- suddivisione della serie storica su diverse dimensioni di aggregazione;
- correzione supervisionata degli outlier;
- implementazione della versione robusta di Holt-Winters.

Abbiamo potuto quindi trarre delle conclusioni, tra queste possiamo citare le seguenti:

- la lunghezza del periodo di training è fondamentale per le prestazioni del sistema (a fronte di un costo di sviluppo minimo è possibile trovare un buon compromesso tra l'aumento della precisione e il costo computazionale);
- la scelta oculata in cui partizionare delle serie storiche migliora la velocità operativa del processo di investigazione e permette di potere attivare misure, volte ad arginare le truffe, più selettive;

- l'implementazione di un sistema che permette la correzione supervisionata dei dati ha permesso di eliminare completamente gli effetti degli outlier qualora si sia consapevoli della natura anomala dell'elemento della serie storica, questo è possibile in quanto il numero degli allarmi da verificare è limitato a pochi casi al giorno;
- l'implementazione della versione robusta della tecnica Holt-Winters, ispirata dall'articolo [GFC10] ha mostrato un buon miglioramento della predizione in presenza di outlier ma ha presentato anche un peggioramento delle predizioni rispetto alla tecnica classica in tutti gli altri casi, questo potrebbe essere dovuto alla mancata modellazione della stagionalità multipla;

Un possibile sviluppo futuro del sistema di rilevamento delle frodi consiste nell'estensione della tecnica Holt-Winters per il supporto di stagionalità multiple.

In conclusione questa tesi ha dimostrato che, con opportuni accorgimenti, è possibile migliorare la qualità delle previsioni dei volumi lordi delle transazioni di un payment service provider nel settore del carrier billing.

Glossario

A/B testing analisi comparativa di PSP dove il merchant divide in parte uguali il traffico degli utenti che desiderano effettuare un acquisto, misurandone il conversion rate e gli introiti. 3

aggregatore azienda partner che permette, implementando un'unica integrazione, di raggiungere più credit provider per effettuare operazioni di billing. 1

API Application Programming Interface. 15

business model La distinzione tra pagamenti *una tantum* (pull), attivazione di pagamenti ricorrenti (subscription activation) e rinnovi (subscription renewal). 48

buyer acquirente, colui che acquista il bene attraverso il sistema di pagamento. 3, 10, 14, 16, 23, 26

codice PIN codice identificativo, talvolta temporaneo il cui scopo è quello di identificare l'utente certificando il possesso del telefono cellulare. 6, 99

conversion rate rapporto tra il numero di persone che aprono la pagina di pagamento e quelle che portano a termine con successo la transazione, o in alternativa la quantità di denaro mediamente percepita per ogni utente che apre la pagina di pagamento. 2, 3, 97

- credit device** identifica l'utenza sulla quale viene addebitata la transazione, può essere un MSISDN oppure l'identificativo della linea telefonica. 1, 26
- credit provider** operatore telefonico o internet service provider, colui che *detiene* il credito telefonico. 1–3, 7, 8, 10, 11, 14, 15, 26, 98
- CSV** comma separated values. 21
- customer base** in italiano clientela, l'insieme degli utenti di un determinato servizio. 2
- DBMS** Database Management System. 16
- delivery report** notifica da parte del credit provider dell'avvenuta (o mancata) consegna di un messaggio. 7
- GTV** gross transaction value. 23, 24, 27, 28, 30, 40
- HTTP** Hypertext Transfer Protocol. 14
- item** il bene acquistato in una purchase. 3, 11
- MAD** deviazione media assoluta. 58
- merchant** L'azienda che mette in vendita il bene, nel caso del carrier billing si parla prevalentemente di *digital merchant*. 1, 3, 6, 7, 10, 11, 14–16, 23, 25, 26, 28, 36, 40, 48, 54, 63, 97
- Mobile Station International Subscriber Directory Number** numero di telefono comprensivo di prefisso internazionale (es. 393401234567). 24, 98
- MSFE** mean squared forecast error. 40, *vedi* MSPE
- MSISDN** numero di telefono. 24, 98, 100, *vedi* Mobile Station International Subscriber Directory Number

MSPE mean squared prediction error. 40, 46–48, 60, 72

opt-in l'azione, compiuta dall'acquirente per esprimere la sua volontà di completare l'acquisto o l'iscrizione al servizio in abbonamento. 4

pagina di pagamento pagina dove l'utente inserisce il numero di telefono e l'eventuale pin per effettuare un pagamento. 3, 6, 7

payment service provider azienda che abilita i merchant all'effettuare pagamenti elettronici. 3

phishing azione fraudolenta volta all'ottenimento di informazioni personali come ad esempio credenziali di accesso a siti Web o bancari mediante richieste in apparenza legittime. 11

PIN Personal Identification Number. 6, *vedi* codice PIN

premium SMS messaggio SMS in entrata o in uscita che comporta l'addebito di un importo sul credito telefonico. 6

PSP Payment Service Provider. III, XI, 1, 3, 6–13, 23, 26, 30, 40, 45, 50, 53, 97, *vedi* payment service provider

purchase un acquisto singolo, non ricorrente. 98

RDBMS Relational Database Management System. 16

REST REpresentational State Transfer. 14

saldo somma algebrica delle transazioni registrate su di un conto. 23

short code numero telefonico speciale, tipicamente a 4-6 cifre dedicato alla gestione del carrier billing. 6, 7

SSE sum of squared errors of prediction. 22, 27, 46

subscription iscrizione ad un servizio in abbonamento. 3

walk in l'atto di inserire il proprio MSISDN nella pagina di pagamento, in alcuni flussi di pagamento questo passaggio avviene automaticamente.
4, 6

Bibliografia

- [Edd13] Dirk Eddelbuettel. *Seamless R and C++ integration with Rcpp*. Springer, 2013.
- [Eva04] Eric Evans. *Domain-driven design: tackling complexity in the heart of software*. Addison-Wesley Professional, 2004.
- [Fie00] Roy Thomas Fielding. *Architectural styles and the design of network-based software architectures*. PhD thesis, University of California, Irvine, 2000.
- [GFC10] Sarah Gelper, Roland Fried, and Christophe Croux. Robust forecasting with exponential and holt-winters smoothing. *Journal of forecasting*, 29(3):285–300, 2010.
- [GKO⁺08] Phillip G Gould, Anne B Koehler, J Keith Ord, Ralph D Snyder, Rob J Hyndman, and Farshid Vahid-Araghi. Forecasting time series with multiple seasonal patterns. *European Journal of Operational Research*, 191(1):207–222, 2008.
- [Hol04] Charles C Holt. Forecasting seasonals and trends by exponentially weighted moving averages. *International journal of forecasting*, 20(1):5–10, 2004.
- [Kal04] Prajakta S Kalekar. Time series forecasting using holt-winters exponential smoothing. *Kanwal Rekhi School of Information Technology*, 4329008:1–13, 2004.

-
- [PLSG10] Clifton Phua, Vincent Lee, Kate Smith, and Ross Gayler. A comprehensive survey of data mining-based fraud detection research. *arXiv preprint arXiv:1009.6119*, 2010.
- [R C13] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013.
- [Tay03] James W Taylor. Short-term electricity demand forecasting using double seasonal exponential smoothing. *Journal of the Operational Research Society*, 54(8):799–805, 2003.
- [Ver13] Vaughn Vernon. *Implementing domain-driven design*. Addison-Wesley, 2013.
- [Win60] Peter R Winters. Forecasting sales by exponentially weighted moving averages. *Management Science*, 6(3):324–342, 1960.
- [ZBLN97] Ciyou Zhu, Richard H Byrd, Peihuang Lu, and Jorge Nocedal. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software (TOMS)*, 23(4):550–560, 1997.